



UNIVERSIDADE FEDERAL DE SANTA CATARINA  
CENTRO DE CIÊNCIAS DA EDUCAÇÃO  
DEPARTAMENTO DE CIÊNCIAS DA INFORMAÇÃO  
CURSO CIÊNCIA DA INFORMAÇÃO

Emanuelle Bavaresco Teodoro

**Estudo de caso do Uso da Metodologia Ágil em um Empresa de Base  
Tecnológica**

Florianópolis

2024

Emanuelle Bavaresco Teodoro

**Estudo de caso do Uso da Metodologia Ágil em um Empresa de Base  
Tecnológica**

Trabalho de Conclusão de Curso submetido ao curso de Ciência da Informação do Centro de Ciências da Educação da Universidade Federal de Santa Catarina como requisito parcial para a obtenção do título de Bacharela em Ciência da Informação.

Orientador(a): Profa. Nathália Berger Werlang,  
Dra.

Florianópolis

2024

Bavaresco Teodoro, Emanuelle  
Estudo de caso do Uso da Metodologia Ágil em um  
Empresa de Base Tecnológica / Emanuelle Bavaresco Teodoro  
; orientadora, Nathália Berger Werlang, 2024.  
64 p.

Trabalho de Conclusão de Curso (graduação) -  
Universidade Federal de Santa Catarina, Centro de Ciências  
da Educação, Graduação em Ciência da Informação,  
Florianópolis, 2024.

Inclui referências.

1. Ciência da Informação. 2. Tecnologia. 3. Metodologia  
ágil. I. Berger Werlang, Nathália . II. Universidade  
Federal de Santa Catarina. Graduação em Ciência da  
Informação. III. Título.

Emanuelle Bavaresco Teodoro

**Estudo de caso do Uso da Metodologia Ágil em um Empresa de Base  
Tecnológica**

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do título de Bacharel e aprovado em sua forma final pelo Curso de Ciência da Informação.

Local Florianópolis, 03 de 12 de 2024.

---

Coordenação do Curso

**Banca examinadora**

---

Profa. Nathália Berger Werlang, Dr.(a)  
Orientador(a)

---

Prof. Ricardo Alexandre Reinaldo de Moraes, Dr.(a)  
Instituição UFSC

---

Prof. Márcio Matias, Dr.(a)  
Instituição UFSC

---

Profa. Rogério da Silva Nunes, Dr.(a)  
Instituição UFSC

Florianópolis, 2024.

## RESUMO

Este trabalho aborda o uso de metodologias ágeis, com foco no framework Scrum, para organização e desenvolvimento de projetos de software, explorando sua aplicação em uma empresa de base tecnológica. A pesquisa compreende uma análise das práticas adotadas na empresa, além de examinar o desempenho das equipes de *front-end* e *back-end* em 16 sprints consecutivos. Os dados coletados revelaram uma preferência consolidada pelas metodologias ágeis. Contudo, a análise revelou a possibilidade de abordagens híbridas, que combinam flexibilidade e estruturação. A análise comparativa entre os times de *front-end* e *back-end* evidenciou diferenças significativas no desempenho. O *front-end* apresentou maior produtividade e velocidade média (27,3 itens concluídos e velocidade média de 53,75), mas com maior variabilidade nos resultados, como evidenciado pelo desvio padrão de 8,5 nos itens concluídos e 13,61 na velocidade. O *back-end* demonstrou desempenho mais estável, mas com menor produtividade (18,5 itens concluídos e velocidade média de 40,88), com desvio padrão de 6,2 nos itens concluídos e 13,87 na velocidade, destacando-se a ausência de documentação como fator possível para inconsistências. Os resultados indicam que o Scrum é eficaz no gerenciamento de projetos em ambientes dinâmicos. Contudo, sua aplicação deve ser adaptada às especificidades do contexto organizacional, reforçando a necessidade de abordagens híbridas como alternativa viável para equilibrar flexibilidade e previsibilidade em diferentes cenários de projetos.

**Palavras-chave:** Metodologias ágeis; Scrum; Desenvolvimento de software; Gestão de projetos; Tecnologia.

## **ABSTRACT**

This work addresses the use of agile methodologies, focusing on the Scrum framework, for organizing and developing software projects, exploring its application in a technology-based company. The research includes an analysis of the practices adopted by the company, as well as the performance of *front-end* and *back-end* teams over 16 consecutive sprints. The collected data revealed a strong preference for agile methodologies. However, the analysis highlighted the potential for hybrid approaches that combine flexibility and structure. The comparative analysis between the *front-end* and *back-end* teams revealed significant performance differences. The front-end team showed higher productivity and average velocity (27.3 completed items and an average velocity of 53.75), but with greater variability in results, as evidenced by a standard deviation of 8.5 in completed items and 13.61 in velocity. The *back-end* team demonstrated more stable performance but lower productivity (18.5 completed items and an average velocity of 40.88), with a standard deviation of 6.2 in completed items and 13.87 in velocity, highlighting the lack of documentation as a possible factor for inconsistencies. The results indicate that Scrum is effective in managing projects in dynamic environments. However, its application should be adapted to the specifics of the organizational context, reinforcing the need for hybrid approaches as a viable alternative to balance flexibility and predictability in different project scenarios.

**Keywords:** Scrum, Agile Methodologies, Software Development, Project Management, Technology.

## LISTA DE FIGURAS

<b>Figura 1</b> - Processo Scrum.....	26
<b>Figura 2</b> - Gráfico Burndown.....	31
<b>Figura 3</b> - Modelo de desenvolvimento Waterfall.....	34
<b>Figura 4</b> - Mapeamento Funções Scrum.....	38
<b>Figura 5</b> - Gráfico itens concluídos por sprint.....	41
<b>Figura 6</b> - Gráfico velocidade por sprint.....	41
<b>Figura 7</b> - Gráfico de barras itens concluídos e não concluídos.....	42
<b>Figura 8</b> - Gráfico itens concluídos por sprint.....	44
<b>Figura 9</b> - Gráfico velocidade por sprint.....	45
<b>Figura 10</b> - Gráfico de barra itens concluídos e itens não concluídos.....	45
<b>Figura 11</b> - Questão 1 do questionário.....	48
<b>Figura 12</b> - Questão 2 do questionário.....	48
<b>Figura 13</b> - Questão 3 do questionário.....	49
<b>Figura 14</b> - Questão 4 do questionário.....	49
<b>Figura 15</b> - Questão 5 do questionário.....	50
<b>Figura 16</b> - Questão 6 do questionário.....	50
<b>Figura 17</b> - Questão 7 do questionário.....	51
<b>Figura 18</b> - Questão 8 do questionário.....	51
<b>Figura 19</b> - Questão 9 do questionário.....	52
<b>Figura 20</b> - Questão 10 do questionário.....	52

## LISTA DE TABELAS

<b>Tabela 2</b> - Métricas Time Front End.....	39
<b>Tabela 3</b> - Estatística Descritiva.....	40
<b>Tabela 4</b> - Métricas Time Back End.....	43
<b>Tabela 5</b> - Análise Descritiva Back End.....	44

## LISTA DE QUADROS

<b>Quadro 1</b> - Os 12 princípios da metodologia ágil.....	20
---	----

## SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>16</b>
1.1 JUSTIFICATIVA.....	17
1.2 OBJETIVOS.....	17
1.2.1 Objetivo Geral.....	18
1.2.2 Objetivos Específicos.....	18
<b>2. FUNDAMENTAÇÃO TEÓRICA.....</b>	<b>18</b>
2.1 GESTÃO DE PROJETOS.....	18
2.2 DEFINIÇÃO DE METODOLOGIA ÁGIL.....	19
2.3 DESAFIOS DO DESENVOLVIMENTO ÁGIL.....	22
2.4 DEFINIÇÃO SCRUM.....	23
2.5 PAPÉIS.....	24
2.5.1 Product Owner.....	24
2.5.2 Scrum Master.....	25
2.6 ATIVIDADES E ARTEFATOS DO SCRUM.....	26
2.6.1 Requisitos e user stories.....	27
2.6.2 Product Backlog.....	27
2.6.3 Sprint backlog.....	28
2.6.4 Planning Poker.....	28
2.6.5 Sprint.....	29
2.6.6 Sprint Planning.....	31
2.6.7 Daily Scrum.....	32
2.6.8 Sprint Review.....	32
2.6.9 Sprint Retrospective.....	32
2.7 Velocidade da Sprint.....	33
2.7 COMPARATIVO ENTRE METODOLOGIAS ÁGEIS E TRADICIONAIS.....	33
<b>3. METODOLOGIA.....</b>	<b>35</b>
3.1 MAPEAMENTO DA EMPRESA.....	35
3.2 COLETA DE DADOS.....	35
3.3 ANÁLISE DE DADOS.....	36
3.3.1 Análise Estatística Descritiva das métricas das equipes.....	36
3.3.2 Análise Estatística Descritiva do Questionário.....	37
<b>4. RESULTADOS.....</b>	<b>38</b>
4.1 CONTEXTO DA EMPRESA.....	38
4.1.1 Mapeamento das Funções com Scrum da Empresa.....	38
4.1.2 Equipe de Desenvolvimento de Software e Metodologia Ágil.....	40

4.2 ANÁLISE DAS MÉTRICAS.....	41
4.2.1. Análise Equipe Front End.....	41
4.2.2 Análise Equipe Back End.....	46
4.3 QUESTIONÁRIO.....	50
<b>5. ANÁLISE COMPARATIVA DOS RESULTADOS.....</b>	<b>54</b>
5.1 COMPARATIVO METODOLOGIA SCRUM E ÁGIL.....	54
5.2 USO DO SCRUM NA EMPRESA.....	54
5.3 COMPARATIVO POR TIPO DE PROJETO(FRONT-END E BACK-END).....	55
<b>6. CONCLUSÃO.....</b>	<b>57</b>

## 1 INTRODUÇÃO

A adoção de metodologias ágeis no desenvolvimento de software tem se intensificado nos últimos anos, refletindo uma tendência crescente em empresas de base tecnológica que buscam otimizar seus processos, reduzir custos e entregar produtos de alta qualidade em prazos reduzidos (Scott et al., 2014). Nesse contexto, surgem duas principais abordagens de gestão: as práticas tradicionais de gerenciamento de projetos e as metodologias ágeis, como o Scrum, que têm ganhado destaque por sua flexibilidade e adaptação a ambientes dinâmicos (Ravaglia et al., 2021).

Inicialmente, as metodologias ágeis foram propostas por um grupo de 17 metodologistas, cujos nomes, formaram a Agile Software Development Alliance, em fevereiro de 2001 (Ambler, 2004). E assim, foi definido um manifesto contendo as melhores práticas para o desenvolvimento de software, incluindo, também, um conjunto de princípios com critérios para os processos de desenvolvimento ágil e 4 valores (Ambler, 2004).

A aplicação correta da metodologia Ágil pode trazer diversos benefícios para as organizações. Entre os principais, destacam-se a troca ampla de conhecimento, o desenvolvimento de habilidades sociais, o feedback constante e a construção de confiança entre os profissionais envolvidos (Solinski; Petersen, 2016). Para os clientes, os benefícios mais visíveis são a entrega de produtos com maior valor agregado, uma melhor relação com os fornecedores e um aumento significativo na qualidade do produto (Solinski; Petersen, 2016). Esses resultados advêm da estreita colaboração entre as partes, o que permite uma compreensão mais clara dos requisitos e a dedicação de mais tempo à resolução de problemas, em vez de apenas à coleta de informações (Larson; Chang, 2016). A metodologia Ágil, portanto, é caracterizada por ciclos curtos de interação, testes contínuos e equipes autogerenciáveis, elementos que contribuem para um desenvolvimento mais eficiente e adaptável (Highsmith, 2002).

Esse modelo de gestão requer monitoramento constante e feedback dos clientes, facilitados por incrementos rápidos e frequentes no desenvolvimento do software, o que diminui os riscos relacionados às incertezas do projeto (Ravaglia et al., 2021).

Entretanto, como destacam Pries-Heje e Baskerville (2017), as práticas do Scrum estão em um estado contínuo de adaptação, o que ilustra a flexibilidade e a dinâmica da metodologia ágil.

Portanto, torna-se relevante a comparação entre a teoria das metodologias ágeis e sua prática em ambientes reais. Enquanto a teoria enfatiza princípios como colaboração, feedback constante e adaptação às mudanças, a prática pode revelar desafios e limitações que não são abordados nos modelos teóricos. Ademais, o estudo comparativo permitirá identificar as lacunas existentes entre o que é preconizado pelas metodologias ágeis e o que realmente ocorre nas empresas que as adotam.

## 1.1 JUSTIFICATIVA

O desenvolvimento de software é um campo em constante evolução, onde a eficiência e a adaptabilidade são essenciais para o sucesso dos projetos. Nesse contexto, as metodologias ágeis têm ganhado destaque, em oposição à visão mecanicista tradicional, por proporcionarem uma abordagem mais flexível (Moe et al., 2009).

Segundo Dingsoyr et al. (2012), os princípios ágeis promovem a entrega contínua de software funcional, envolvendo os clientes no processo de desenvolvimento para garantir resultados mais satisfatórios através de feedbacks.

Nesse sentido, um estudo de caso desse fenômeno pode oferecer sua melhor compreensão, a fim da otimização dos processos de desenvolvimento de software, proporcionando um entendimento mais claro das dinâmicas informacionais que ocorrem em ambientes ágeis.

## 1.2 OBJETIVOS

Esta seção apresenta os objetivos do estudo, incluindo o objetivo geral e os objetivos específicos que orientam a pesquisa.

### 1.2.1 Objetivo Geral

Este estudo tem como objetivo geral analisar a aplicação da metodologia ágil Scrum em uma empresa de base tecnológica, com foco no desempenho das equipes de desenvolvimento de software (*front-end* e *back-end*) e na eficiência dos processos de entrega de software.

### 1.2.2 Objetivos Específicos

1. Analisar a performance das equipes de *front-end* e *back-end* na utilização do Scrum, identificando padrões e variações ao longo do tempo.
2. Explorar as percepções e experiências dos colaboradores sobre a eficácia da metodologia ágil, com base em dados coletados por questionário.
3. Mapear o funcionamento da empresa relacionado ao uso da metodologia Scrum, incluindo funções, papéis e práticas adotadas pelas equipes.

## 2. FUNDAMENTAÇÃO TEÓRICA

Esta seção apresenta os conceitos e teorias que embasam o presente estudo. O objetivo é fornecer uma base teórica sólida que sustente a análise e os resultados obtidos, destacando as principais abordagens, ferramentas e métodos relacionados ao tema.

### 2.1 GESTÃO DE PROJETOS

A gestão de projetos é uma prática indispensável no desenvolvimento de software, pois oferece a estrutura necessária para planejar, organizar e executar iniciativas com eficiência. Historicamente, diversas soluções de software fracassaram antes de serem concluídas, muitas vezes devido à ausência de um planejamento estruturado e da utilização de ferramentas apropriadas para gerenciar os projetos (Hayat et al., 2019).

Assim, a gestão de projetos de software envolve um conjunto de áreas de conhecimento, ferramentas e técnicas específicas. Ela abrange dez áreas de conhecimento, sendo quatro principais e seis complementares. As áreas principais incluem a gestão de custos do projeto, escopo e cronograma, enquanto as complementares englobam a gestão de recursos humanos, aquisições, comunicações, partes interessadas, riscos e integração do projeto. Estas últimas atuam como elementos integradores, promovendo a coesão e a eficácia dos processos de gestão (Hossain; Emam, 2010).

Além disso, a gestão de projetos é estruturada em cinco processos: iniciação, planejamento, execução, monitoramento e controle, e encerramento. Cada uma dessas etapas desempenha um papel específico na organização e condução dos projetos, permitindo o alcance dos objetivos estabelecidos e a mitigação de riscos ao longo do ciclo de vida do software. (Hossain; Emam, 2010)

### 2.2 DEFINIÇÃO DE METODOLOGIA ÁGIL

A partir de meados da década de 1990, as metodologias ágeis e suas práticas emergiram como uma resposta formal às frequentes mudanças nos requisitos de software e nas expectativas dos clientes. (Williams, 2010)

As práticas ágeis, incluem a comunicação face a face, reuniões de planejamento e retrospectivas facilitadas por equipes autogerenciáveis e multifuncionais, promovem uma integração contínua com testes.(Solinski; Petersen, 2016). Esses elementos são fundamentais para permitir entregas rápidas e de alta qualidade, com um foco constante na adaptação e no atendimento às necessidades do cliente (Solinski; Petersen, 2016).Um dos principais princípios das metodologias ágeis é a preferência por iterações curtas de desenvolvimento, que permitem um feedback contínuo e a adaptação de requisitos. (Williams,2010)

A "Manifesto Ágil", criado por um grupo de 17 consultores de engenharia de software em 2001, delinearam os princípios centrais do manifesto, conforme apontado no Quadro 1:

**Quadro 1:** Os 12 princípios da metodologia ágil

<b>Princípio</b>	<b>Descrição</b>
Princípio 1	A prioridade máxima é entregar software de qualidade de forma contínua e antecipada.
Princípio 2	Mudanças nos requisitos são bem-vindas, mesmo nas fases finais, para beneficiar o cliente.
Princípio 3	O software funcional deve ser entregue com frequência, preferencialmente em períodos curtos.
Princípio 4	Empresários e desenvolvedores devem colaborar diariamente ao longo do projeto.
Princípio 5	Projetos devem ser construídos em torno de indivíduos motivados, fornecendo o suporte necessário e confiando em sua capacidade.
Princípio 6	A comunicação mais eficaz é a conversa cara a cara dentro da equipe.
Princípio 7	O progresso é medido principalmente pela entrega de software funcional.
Princípio 8	Todos os envolvidos devem manter um ritmo constante e sustentável indefinidamente.
Princípio 9	A atenção à excelência técnica e ao bom design aumenta a agilidade do desenvolvimento.
Princípio 10	A simplicidade, ou a arte de maximizar o trabalho não feito, é fundamental.
Princípio 11	As melhores soluções surgem de equipes que se auto-organizam.
Princípio 12	A equipe deve refletir regularmente sobre sua eficácia e ajustar seu comportamento conforme necessário.

**Fonte:** Adaptação da autora a partir de BECK et al. (2001)

Ademais, as metodologias ágeis baseiam-se em quatro valores fundamentais, definidos por Beck et al. (2001), que orientam as equipes ao longo do desenvolvimento de projetos de software. O primeiro valor, “indivíduos e interações sobre processos e ferramentas”, enfatiza a relevância do fator humano no desenvolvimento. A interação e comunicação efetiva entre os membros da equipe são priorizadas em relação à formalização excessiva dos processos, que pode restringir a criatividade e flexibilidade, essenciais para inovação e resolução eficaz de problemas (Beck et al., 2001).

O segundo valor, “software em funcionamento sobre documentação abrangente”, promove o foco no desenvolvimento de versões utilizáveis do produto, em vez de dedicar tempo excessivo à produção de documentação detalhada. Esse enfoque facilita a entrega rápida e o feedback contínuo, permitindo que a equipe aprimore o produto com agilidade, mantendo o desenvolvimento do software como prioridade principal (Beck et al., 2001).

O terceiro valor, “colaboração com o cliente sobre negociação de contrato”, desafia os métodos tradicionais que tendem a fixar os requisitos logo no início do projeto por meio de contratos rígidos. Em vez disso, as metodologias ágeis incentivam a colaboração constante com o cliente, permitindo que a equipe ajuste o produto em conformidade com as reais necessidades do usuário à medida que o software evolui, criando um processo mais dinâmico e centrado no cliente (Beck et al., 2001).

Por fim, o quarto valor, “responder a mudanças sobre seguir um plano”, destaca a flexibilidade das metodologias ágeis em aceitar modificações nos requisitos, mesmo em fases avançadas do desenvolvimento. Esse valor possibilita o ajuste contínuo do planejamento a cada iteração, o que diferencia as metodologias ágeis das abordagens tradicionais, ao priorizar a adaptação eficaz a novas informações e necessidades dos clientes em vez de aderir rigidamente a um plano fixo (Beck et al., 2001).

## 2.3 DESAFIOS DO DESENVOLVIMENTO ÁGIL

O entendimento dos desafios enfrentados no desenvolvimento ágil de software é fundamental para que as organizações possam elaborar melhores planos e estimativas, a fim de mitigar ou lidar com esses obstáculos, garantindo, assim, uma maior probabilidade de sucesso na execução dos projetos. A identificação e compreensão desses desafios permitem que as equipes de desenvolvimento possam adotar estratégias apropriadas para enfrentá-los, minimizando seus impactos negativos sobre a qualidade e o desempenho do projeto (Fitriani et al., 2024).

De acordo com os estudos analisados, foram identificados 30 desafios significativos na implementação da metodologia ágil. A Tabela a seguir, presente no artigo, resume os temas de desafio derivados dos 20 estudos selecionados, com a frequência de sua ocorrência e os respectivos artigos relacionados. Entre os desafios mais comuns, destacam-se a gestão de equipes e a distribuição de equipes, mencionados em 9 dos 20 artigos (45%). Esses desafios surgem devido às dificuldades em coordenar as atividades e manter a comunicação eficaz entre os membros da equipe, especialmente quando distribuídos geograficamente. (Fitriani et al., 2024)

Em seguida, destacam-se os desafios de priorização de requisitos, documentação, mudanças e ampliação de requisitos, que aparecem em 30% dos estudos analisados. A constante necessidade de ajustar os requisitos do projeto e a documentação que acompanha o processo pode resultar em problemas de comunicação e controle de qualidade. Além disso, a organização, os processos e o monitoramento de progresso, juntamente com o feedback, são desafios abordados em 25% dos estudos, sendo cruciais para garantir que as entregas ocorram conforme o planejado e que os stakeholders estejam constantemente atualizados sobre o andamento do projeto. Outros desafios, embora identificados, foram considerados menos significativos, pois aparecem em menos de 25% dos estudos, incluindo problemas de integração de equipes multidisciplinares e dificuldades na gestão de *stakeholders* (Fitriani et al., 2024).

## 2.4 DEFINIÇÃO SCRUM

Segundo Ribeiro et al (2018) o ambiente de negócios atual é amplamente caracterizado por instabilidade, incerteza, complexidade e ambiguidade, fatores que geram um cenário denominado VUCA (Volátil, Incerto, Complexo e Ambíguo). Em resposta a esse contexto, surgiu a metodologia Ágil. Em que a agilidade pode ser definida como a capacidade da equipe de projeto de responder rapidamente a mudanças e lidar com os impactos causados por fatores internos e externos(Pereira e Russo, 2018). Essa capacidade é influenciada pela organização em sua habilidade de gerenciar mudanças e a adaptação contínua ao ambiente dinâmico (Conforto et al., 2016).

O termo “*Scrum*”, tem sua origem em um termo do rugby, onde se refere à forma de reiniciar a partida após uma interrupção (Rubin, 2017). Sendo que a história do Scrum remonta ao artigo “The New New Product Development Game”, publicado por Takeuchi e Nonaka na Harvard Business Review em 1986. Nesse trabalho, os autores descreveram práticas adotadas por empresas como Honda, Canon e Fuji-Xerox, que utilizaram uma abordagem escalável fundamentada em equipes para o desenvolvimento de produtos, denominada *all-at-once* (Rubin, 2017). Segundo Rubin (2017) o artigo desempenhou um papel fundamental na consolidação dos conceitos que posteriormente seriam formalizados no Scrum.

Nesse sentido, caracteriza-se por ser:

O Scrum trata, principalmente, do nível da equipe. Ele permite que as pessoas trabalhem juntas de forma eficaz e, ao fazer isso, possibilita que elas produzam produtos complexos e sofisticados. O Scrum é uma espécie de engenharia social que busca alcançar a satisfação de todos os envolvidos, promovendo a cooperação. A cooperação surge à medida que as equipes se auto-organizam em incubadoras fomentadas pela gestão. Usando o Scrum, as equipes desenvolvem produtos de forma incremental e empírica. Elas são guiadas pelo seu conhecimento e experiência, em vez de por planos de projeto definidos formalmente (Schwaber; Beedle, 2001, p. 2).

No Scrum “*Sprint*” é um conceito central da metodologia, representando um ciclo de desenvolvimento com duração fixa, também chamado de iteração.(Williams, 2010)

Contudo, vale destacar que, de acordo com Singh (2008), o Scrum tradicional, embora eficaz na gestão de projetos, enfrenta desafios quando se trata de integrar aspectos como a usabilidade do produto. Em muitos casos, a usabilidade é considerada uma reflexão posterior, o que pode impactar negativamente a experiência do usuário. Como observou Singh (2008), o Scrum não enfatiza o desenvolvimento de protótipos ou a validação de requisitos com os usuários antes do início do desenvolvimento. Isso pode levar a um foco maior na conclusão do trabalho em vez de aprimorar a experiência do usuário, o que representa uma limitação significativa da metodologia tradicional.

Singh (2008) também destaca que algumas organizações tentam contornar essa limitação incorporando equipes de usabilidade que trabalham à frente do ciclo de desenvolvimento. Essas equipes fornecem aos desenvolvedores artefatos como wireframes, ou realizam testes de aceitação de usabilidade para as user stories, mas o foco central continua sendo a entrega do produto, em vez da experiência do usuário.

## 2.5 PAPÉIS

O framework scrum não delimita todos os papéis que devem existir em uma equipe ágil, mas sim, requer os essenciais: *product owner*, *product master* e a equipe de desenvolvimento (Rubin, 2017).

### 2.5.1 Product Owner

De acordo com Pócsová et al. (2020), o Product Owner atua na definição da visão do projeto ou produto, sendo responsável pela comunicação clara e transparente com a equipe, clientes e sociedade, bem como, "especificar e reunir histórias de usuário individuais e seus critérios de aceitação." (Singh, 2008, p.555). Além disso, é ele quem define a ordem e as funcionalidades que devem ser desenvolvidas e gerencia o *product backlog* (Rubin, 2017).

Singh (2008) aponta que o Scrum não cobre adequadamente as necessidades de usabilidade, tanto na criação inicial das user stories quanto nas modificações incrementais decorrentes do *feedback* dos usuários. Esse problema deve-se, em parte, ao fato de que os *product owners* estão frequentemente focados em demandas de marketing e vendas, o que compromete a atenção à usabilidade. Além disso, metodologias ágeis tradicionais raramente permitem o desenvolvimento de uma “visão de experiência do usuário”, essencial para assegurar uma experiência coesa e consistente (Ferreira; Noble; Biddle, 2007).

### **2.5.2 Scrum Master**

O Scrum Master atua como um facilitador, conduzindo a equipe ao longo do processo de construção do produto por meio dos valores, princípios e práticas do Scrum. Ele é responsável por manter a equipe protegida de influências externas e gerenciar impedimentos que possam afetar a produtividade (Rubin, 2017). Além disso, ele auxilia na conquista dos objetivos do projeto ao proporcionar um ambiente de trabalho produtivo e garantir que a equipe siga os valores ágeis e os métodos do Scrum (Pócsová et al., 2020).

Contudo, conforme destacado por Rubin (2017), o Scrum Master não exerce controle sobre a equipe, atuando mais como um líder do que como um gerente.

### **2.5.3 Equipe de Desenvolvimento**

Segundo Pócsová et al. (2020), a equipe de desenvolvimento é responsável por entregar um produto potencialmente comercializável ao final de cada sprint, conforme a *Definition of Done*. A equipe organiza suas atividades de forma autônoma e, assim, os membros devem transformar os requisitos do usuário em um produto funcional (Pócsová, et al., 2020).

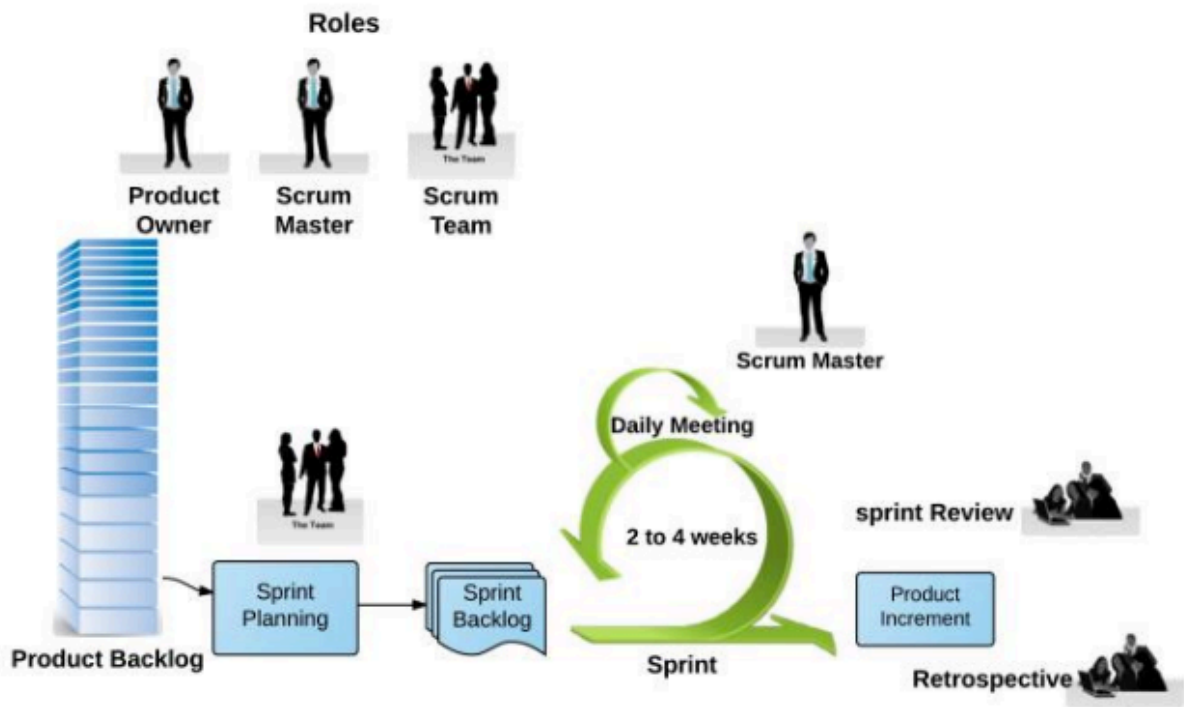
De modo geral, a equipe de desenvolvimento é composta de 5 a 9 pessoas, sendo que os membros possuem todas as habilidades para produzir um software funcional e de boa qualidade. (Rubin, 2017).

## 2.6 ATIVIDADES E ARTEFATOS DO SCRUM

As atividades e artefatos formam a estrutura que possibilita a criação de produtos de maneira eficiente e previsível. (Rubin, 2017) As principais atividades no Scrum incluem a Sprint Planning, a Daily Scrum, a Sprint Review e a Sprint Retrospective. Essas reuniões ou cerimônias são organizadas para facilitar a comunicação e o alinhamento do time, garantindo que todos estejam cientes das metas e possam ajustar o curso de acordo com o progresso. Na Sprint Planning, ocorre a definição do escopo para a sprint, enquanto a Daily Scrum é uma breve reunião diária para acompanhar o progresso e identificar impedimentos. A Sprint Review permite a inspeção do incremento do produto, possibilitando o feedback das partes interessadas, enquanto a Sprint Retrospective fecha o ciclo da sprint, incentivando a equipe a refletir sobre melhorias no processo. (Rubin, 2017)

Em relação aos artefatos, o SCRUM organiza-se em três artefatos essenciais: o Product Backlog, uma lista ordenada dos itens priorizados para alcançar o objetivo do projeto, sendo constantemente atualizado e priorizado pelo Product Owner (Rubin, 2017). A Figura 1 apresenta o *framework* Scrum, destacando os papéis, artefatos e eventos principais no processo.

**Figura 1:** Processo Scrum



Fonte: Lopez-Martinez et al., p.142 (2016)

Na parte superior, estão as funções principais: Product Owner, Scrum Master e o Scrum Team. Na base, o Product Backlog alimenta o planejamento da sprint (Sprint Planning), resultando no Sprint Backlog. Durante o ciclo de 2 a 4 semanas, o Sprint é acompanhado por reuniões diárias (Daily Meetings). Ao final, realiza-se a Sprint Review, onde o Incremento do Produto é inspecionado, seguida pela Retrospective, para melhoria contínua do processo.

### 2.6.1 Requisitos e user stories

No Scrum, a gestão de requisitos adota uma abordagem dinâmica e adaptativa, reconhecendo que nem todos os requisitos precisam estar no mesmo nível de detalhe em todas as etapas do desenvolvimento (Rubin, 2017). Requisitos que serão trabalhados em breve devem ser menores e mais detalhados, enquanto aqueles que ainda não estão no foco imediato permanecem maiores e com menos detalhes. Essa abordagem é conhecida como refinamento progressivo, onde

grandes conjuntos de requisitos levemente detalhados são desagregados, de forma *just in time*, em itens menores e mais específicos, à medida que se aproximam da execução.(Rubin, 2017).

As user stories surgem como uma ferramenta eficaz para capturar o valor de negócio desejado em muitos tipos de itens do *product backlog*, especialmente em funcionalidades. Essas histórias são projetadas para serem compreensíveis tanto para profissionais de negócios quanto para técnicos, utilizando uma estrutura simples que serve como um ponto de partida para conversas mais detalhadas. Além disso, as user stories são versáteis, podendo ser escritas em diferentes níveis de granularidade e refinadas progressivamente ao longo do projeto.(Rubin, 2017).

### **2.6.2 Product Backlog**

O Product Backlog é uma lista ordenada dos itens priorizados necessários para atingir o objetivo do produto ou projeto(Rubin, 2017). A ordenação dos trabalhos inicia pelo de maior valor primeiro. Nesse sentido, quem define a ordem é o Product Owner com as informações da equipe scrum e de stakeholders. Essa organização gera uma lista priorizada e é chamada de Product Backlog.(Rubin, 2017) Os itens do Product Backlog são as features necessárias para atender à visão do Product Owner sobre o produto que deve ser criado.(Rubin, 2017)

Para definir os itens, o Product Owner colabora com stakeholders internos e externos, podendo estar em constante evolução à medida que as condições de negócio mudem, ou que o entendimento da equipe sobre o produto aumente. (Rubin, 2017)

No Scrum, o tamanho de cada item do *product backlog* é frequentemente estimado com base em story points ou ideal days. Os story points são uma medida abstrata que reflete a complexidade, o esforço e a incerteza de cada tarefa, permitindo à equipe avaliar o trabalho de forma relativa, sem vincular a estimativa ao tempo real. Já os ideal days representam o número de dias de trabalho ininterrupto que a equipe acredita ser necessário para concluir um item. (Rubin, 2017)

Além disso, chama-se grooming o processo de criação, refinamento, atribuição de estimativas e ordenação dos itens(Rubin 2017). Esse processo envolve

o desdobramento dos requisitos dos clientes, obtidos junto aos stakeholders, em tarefas específicas para serem executadas pela equipe em um ciclo de trabalho.(Ribeiro et al., 2018)

### **2.6.3 Sprint backlog**

O *sprint backlog* é o conjunto de tarefas e itens selecionados do *product backlog* que a equipe de desenvolvimento se compromete a concluir durante uma sprint. O *sprint backlog* é atualizado pela equipe ao longo da sprint para refletir o progresso e é essencial para manter o foco no objetivo da sprint, promovendo transparência e acompanhamento do trabalho em andamento.(Rubin, 2017)

### **2.6.4 Planning Poker**

O Planning Poker é uma técnica utilizada para estimar o esforço necessário para implementar funcionalidades em um projeto, jogado durante a reunião de Sprint Planning(Williams, 2010). Essa prática foi descrita pela primeira vez por James Grenning(2002) e posteriormente popularizada por Mike Cohn em 2002(Rubin, 2017).

Para isso, os itens do Product Backlog são discutidos em equipe, onde cada membro avalia o trabalho necessário para atender aos requisitos. O intuito é garantir que todos os participantes se sintam capacitados para estimar o esforço envolvido. Cada um faz sua estimativa de forma independente, utilizando unidades chamadas "pontos de história". Em seguida, aqueles que apresentaram as estimativas e a partir dessa troca de opiniões, a equipe discute até que todos estejam preparados para uma nova votação sobre as estimativas. Esse processo pode ser repetido várias vezes até que se chegue a um consenso sobre a quantidade de pontos de história atribuídos a cada requisito. Os pontos de história são uma medida relativa e não possuem uma unidade fixa, baseando-se em experiências anteriores com requisitos já finalizados (Williams, 2010).

### 2.6.5 Sprint

A sprint, no Scrum, é definida como uma iteração, ou seja, um ciclo de trabalho estruturado e limitado em tempo, cujo objetivo é entregar um incremento de valor tangível para o cliente. Os sprints possuem duração curta e constante, geralmente variando entre uma semana e um mês, permitindo previsibilidade e cadência ao processo. Embora a consistência no tamanho das sprints seja recomendada, exceções podem ser aplicadas em circunstâncias específicas. Durante uma sprint, mudanças que afetem a composição da equipe ou o escopo do objetivo são desencorajadas, pois comprometeriam a estabilidade e o foco necessário para a execução. Ao final de cada sprint, é produzido um incremento potencialmente entregável do produto, que atende aos critérios estabelecidos na definição de pronto (*definition of done*), previamente acordada pela equipe. Esse processo assegura que cada ciclo agregue valor e mantenha a qualidade alinhada às expectativas do cliente e do mercado. (Rubin, 2017)

Sendo que a sprint ocorre dentro de um *timebox*, ou seja, é o seu tempo de início e fim. (Rubin, 2017) Para isso, ferramentas como o Backlog da Sprint e o *burndown da sprint*, são utilizadas para comunicar o progresso a todas as partes envolvidas. (Albero Pomar et al., 2014)

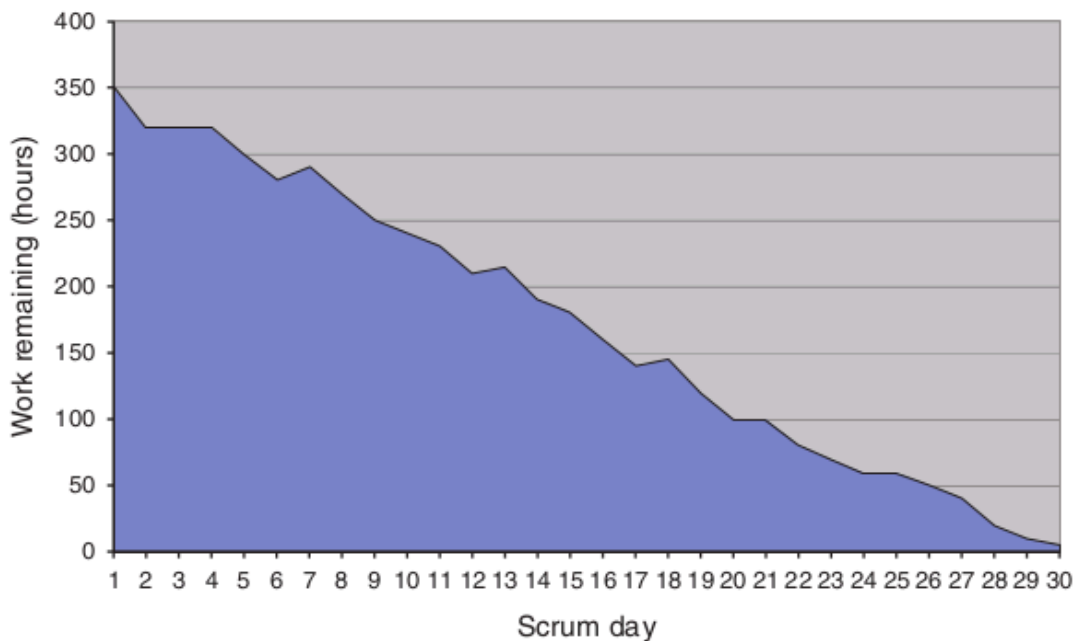
No Scrum, o conceito de *"timebox"* desempenha um papel essencial na gestão do trabalho em processo. Dentro desse período delimitado, a equipe deve trabalhar em um ritmo sustentável para completar o conjunto de tarefas selecionadas, alinhadas ao objetivo da sprint (*sprint goal*). O *timebox* é uma técnica que limita a quantidade de *WIP*, ou seja, o inventário de trabalho que foi iniciado, mas ainda não concluído. Ao planejar cuidadosamente as tarefas a serem realizadas em cada sprint, a equipe compromete-se apenas com itens que acredita poder iniciar e finalizar dentro do ciclo estipulado. Dessa forma, o *timebox* atua como um mecanismo regulador, evitando sobrecarga, promovendo foco e garantindo entregas consistentes e alinhadas ao planejamento. Além de que o *timebox* força a priorizar e realizar a menor quantidade de trabalho que importa mais (Rubin, 2017).

A cada sprint devem ser definidos os *sprint goals*, que descreve o valor e o propósito de negócio de cada sprint (Rubin, 2017). Os objetivos da Sprint são fixos durante sua execução, porém, a cada incremento, o Product Owner tem a

possibilidade de adicionar novas funcionalidades ao projeto que não haviam sido previstas anteriormente.(Schwaber, 2004)

Para monitorar o andamento de um projeto ao longo do tempo, usa-se o gráfico de burndown. Segundo Williams (2004, p. 42), "um gráfico de burndown é uma representação gráfica do trabalho restante em relação ao tempo[...]. O trabalho pendente (ou backlog) normalmente está no eixo vertical e o tempo no eixo horizontal". Essa representação visual permite que a equipe e as partes interessadas acompanhem de forma clara a quantidade de trabalho que ainda precisa ser realizado, à medida que o tempo avança e as tarefas são concluídas, conforme o ilustrado na Figura 2:

**Figura 2:** Gráfico Burndown



**Fonte:** Williams, 2010, p.20.

Srivastava, Bhardwaj e Saraswat (2017) explicam que os gráficos de burndown da Sprint, do produto e de lançamento são ferramentas usadas para acompanhar o progresso e evitar o burnout. Esses gráficos começam com o número máximo de pontos e, ao longo do tempo, esse valor diminui até chegar a zero (Srivastava; Bhardwaj; Saraswat, 2017). Assim, eles ajudam as equipes a monitorar

o progresso durante uma Sprint, permitindo uma visualização clara do trabalho restante em relação ao tempo disponível.

### **2.6.6 Sprint Planning**

A sprint planning é o planejamento da próxima sprint determinado pelo scrum master, product owner e a equipe de desenvolvimento, em que, é determinado o que ela deve alcançar, chamado de sprint goal.(Rubin, 2017). Segundo Lopez-Martinez et al., (2016), a Sprint, possui uma duração de 2 a 4 semanas, é o elemento central do processo, iniciando sempre com uma reunião de planejamento, a Sprint Planning.

Tendo-se o sprint goal, a equipe de desenvolvimento revisa o product backlog e determina os itens de mais alta prioridade que poderão ser atingidos de forma realística,também fornecem uma estimativa do esforço para cada tarefa, geralmente em horas (Rubin, 2017).

### **2.6.7 Daily Scrum**

Ao longo da sprint, realiza-se a reunião diária do Scrum, chamada de Daily Scrum, com o objetivo de identificar problemas e compartilhar o progresso de cada membro da equipe em relação às suas tarefas(Pomar et al., 2014). A reunião dura cerca de 15 minutos onde participam equipe de desenvolvimento e scrum master como facilitador , com o objetivo de dar uma visão geral de como está se dando o desempenho de cada um dentro da sprint rumo ao objetivo final.(Rubin, 2017)

As questões que são abordadas são: “O que eu realizei desde a última daily scrum?”; “O que eu planejo trabalhar para o próximo daily scrum?”; “Quais são os obstáculos ou impedimentos que estão evitando que eu progrida?”(Alam et al., 2017).Vale destacar que a daily scrum não visa solucionar os problemas, nem é uma reunião de status. Mas sim, o sentido geral, é ser uma atividade adaptativa de inspeção e sincronização a fim de que a equipe se auto-organize(Rubin, 2017).

### **2.6.8 Sprint Review**

É a penúltima atividade de inspeção e adaptação, é o momento de comunicar uma revisão das features que acabaram de ser feitas. Os participantes incluem: equipe Scrum, os stakeholders, os sponsors, os clientes e os membros interessados de outras equipes. Espera-se que o resultado seja um fluxo bidirecional de informações, representando uma oportunidade para inspecionar e adaptar o produto(Rubin, 2017).

### **2.6.9 Sprint Retrospective**

A última atividade de inspeção e adaptação é a Sprint Retrospective e ocorre depois da sprint review e antes da próxima sprint planning.(Rubin, 2017) Diferente da sprint review, e sprint retrospective visa inspecionar e adaptar o processo, em vez do produto. O foco é ajustar e melhorar a equipe scrum, e identificar as ações que deverão ser feitas na próxima sprint a fim de alcançar a excelência no seu processo(Werewka; Spiechowicz, 2017). Para Sprints com duração de um mês, estima-se que a retrospectiva deve ter uma duração de até três horas.(Lapkin et al., 2008)

Os autores DERBY et al (2006) sugerem dividir a retrospectiva da sprint em cinco fases: Definir o contexto, Coletar dados, Gerar percepções através de brainstorming, Decidir o que fazer, priorizando os fatores mais importantes, com o exercício "Planning Game" por fim Fechar a retrospectiva reunindo feedback dos participantes(Werewka; Spiechowicz, 2017).

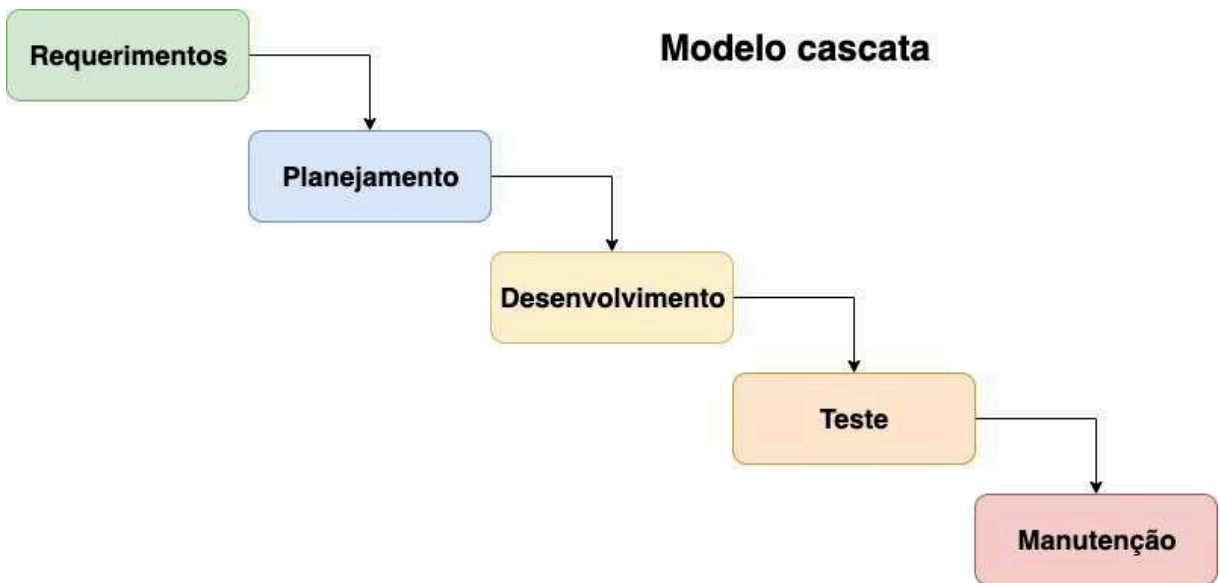
### **2.6.10 Velocidade da Sprint**

A velocidade é um ponto essencial para o planejamento de uma Sprint.(RUBIN, 2017) Segundo Rubin (2017) a velocidade é definida como a quantidade de trabalho completada a cada Sprint e é obtida somando-se os tamanhos dos itens do Product Backlog que foram concluídos no final da Sprint. Um

exemplo seria “A equipe é tipicamente capaz de completar entre 20 e 25 pontos a cada Sprint.” Com um intervalo é possível prever “quando irá ficar pronto”, “quantos itens podemos completar” e “quanto vai custar”(RUBIN, 2017)

## 2.7 COMPARATIVO ENTRE METODOLOGIAS ÁGEIS E TRADICIONAIS

As metodologias tradicionais de desenvolvimento de software, conforme descritas por Rubin (2017), são caracterizadas por uma abordagem sequencial e orientada por planejamento. Esse paradigma busca estruturar o desenvolvimento em fases bem definidas, com o objetivo de antecipar e planejar todas as funcionalidades que o usuário final possa desejar no produto. Um exemplo amplamente conhecido dessa classe de processos é o modelo em cascata (waterfall), que representa uma forma de desenvolvimento prescritivo, Conforme o ilustrado na Figura 3:

**Figura 3:** Modelo de desenvolvimento Waterfall

**Fonte:** Dicionariotec (2022).

De acordo com Rubin (2017), os processos orientados a planejamento assumem que quanto mais detalhado e abrangente for o planejamento inicial, melhor será o entendimento do sistema a ser construído e, conseqüentemente, mais eficiente será sua execução. Esses processos seguem etapas sequenciais que envolvem a realização de uma análise completa de requisitos, seguida por um design detalhado, a construção/codificação, e, por fim, os testes. Essa linearidade caracteriza o modelo sequencial, que é indicado para problemas bem definidos, previsíveis e com baixa probabilidade de mudanças significativas.

No entanto, essa abordagem enfrenta desafios em contextos de desenvolvimento que apresentam incertezas e alta imprevisibilidade, especialmente nas fases iniciais do projeto. O desenvolvimento orientado a planejamento, embora eficiente em cenários estáveis, não é ideal para lidar com as complexidades e mudanças frequentes que caracterizam muitos projetos de software. Nesses casos, metodologias mais adaptativas, como o Scrum, oferecem maior flexibilidade por se basearem em princípios que permitem ajustes contínuos durante o desenvolvimento (Rubin, 2017).

### 3. METODOLOGIA

A presente pesquisa será conduzida utilizando a abordagem de Estudo de Caso, conforme os princípios metodológicos propostos por García-Moreno e Martínez-García (2012). Esta abordagem permite uma análise do uso da metodologia ágil em uma empresa de base tecnológica.

#### 3.1 MAPEAMENTO DA EMPRESA

O primeiro passo consiste na caracterização da situação da empresa em estudo. Foi realizado um levantamento detalhado dos processos existentes, com foco na aplicação das metodologias ágeis. Para isso, foi realizada a documentação dos processos, que incluiu os fluxos de trabalho.

#### 3.2 COLETA DE DADOS

A coleta de dados foi realizada por meio de duas abordagens complementares: a análise de registros na ferramenta Jira e a aplicação de um questionário online estruturado.

A empresa em questão utiliza o Jira como principal ferramenta de gerenciamento de projetos ágeis, juntamente com o Confluence para a documentação de processos e requisitos. No entanto, observou-se que o time de backend não possui uma documentação bem estabelecida no Confluence, o que pode impactar diretamente a clareza e a consistência das informações entre os membros da equipe.

Os dados extraídos do Jira incluem métricas como tempo de entrega, volume de tarefas concluídas, tarefas não concluídas e velocidade (pontos concluídos por sprint). Esses dados foram coletados ao longo de oito semanas, abrangendo 16 sprints, e oferecem uma base objetiva para avaliar o desempenho das equipes de frontend e backend.

Além disso, um questionário online foi aplicado para coletar dados qualitativos e quantitativos sobre a percepção dos colaboradores em relação à eficácia da

metodologia ágil. A população-alvo do questionário foi composta por 12 membros das equipes de desenvolvimento, dos quais sete responderam, resultando em uma taxa de resposta de aproximadamente 58%.

### 3.3 ANÁLISE DE DADOS

A análise dos dados será dividida em duas etapas principais: análise descritiva das métricas das equipes e análise descritiva do questionário.

#### 3.3.1 Análise Estatística Descritiva das métricas das equipes

A análise dos dados constituirá uma análise estatística descritiva das métricas das equipes, utilizando dados extraídos do Jira ao longo de 8 semanas, correspondentes a 16 sprints. Foram consideradas três métricas principais: itens concluídos, itens não concluídos e velocidade, esta última representando o número de pontos concluídos por sprint.

Para cada uma dessas métricas, serão calculadas medidas estatísticas descritivas abordadas por Morettin e Bussab (2010):

- Média aritmética: fornece uma medida de tendência central, representando o valor típico dos dados.
- Desvio padrão: quantifica a dispersão dos dados em relação à média, indicando a variabilidade do desempenho.
- Valores mínimo e máximo: identificam os extremos de desempenho das equipes, revelando os melhores e os piores resultados observados.
- Mediana: representa o ponto central dos dados, sendo uma medida robusta contra a influência de valores atípicos.

De acordo com Morettin e Bussab (2010), essas estatísticas descritivas são fundamentais para compreender o comportamento dos dados e suas tendências principais. Adicionalmente, serão criadas visualizações gráficas para complementar a análise numérica, facilitando a interpretação e comunicação dos resultados:

- Gráfico de barras: será utilizado para ilustrar os itens concluídos e não concluídos por sprint, permitindo a análise do desempenho por período.
- Gráficos de linhas: dois gráficos serão gerados para representar a variação da velocidade ao longo dos sprints e os itens concluídos, destacando mudanças no ritmo de trabalho.

Os resultados obtidos serão comparados entre as equipes de backend e frontend, com o objetivo de identificar padrões, semelhanças e diferenças nos comportamentos observados. A comparação permitirá avaliar a evolução do desempenho das equipes ao longo do período, contribuindo para a análise da eficácia do framework Scrum na organização estudada.

### **3.3.2 Análise Estatística Descritiva do Questionário**

Os dados obtidos por meio do questionário serão analisados utilizando técnicas de estatística descritiva. Essa análise permitirá resumir e descrever as características dos dados coletados, fornecendo uma visão geral das percepções dos colaboradores em relação ao uso da metodologia ágil.

A análise do questionário será conduzida com o objetivo de compreender as percepções e preferências dos respondentes em relação ao uso de metodologias ágeis no desenvolvimento de software, bem como identificar os benefícios, desafios e ajustes necessários para sua aplicação em diferentes contextos.

Inicialmente, as respostas foram organizadas de acordo com aspectos como preferências metodológicas, benefícios percebidos, dificuldades enfrentadas e formas de organização do trabalho. Posteriormente, foi realizada uma verificação de consistência, comparando as respostas iniciais com as subsequentes, a fim de identificar possíveis contradições entre a preferência declarada e a familiaridade prática com as metodologias ágeis.

Essa abordagem, fundamentada em métodos quantitativos e qualitativos, garantiu que as percepções e experiências relatadas pelos respondentes fossem analisadas de forma completa e alinhada aos objetivos do estudo, permitindo

identificar tanto as vantagens amplamente reconhecidas das metodologias ágeis quanto os desafios de sua aplicação em diferentes ambientes organizacionais.

## 4. RESULTADOS

Serão apresentados os resultados obtidos a partir da análise dos dados coletados durante os 16 sprints avaliados. As informações incluem métricas, relacionadas à produtividade e desempenho das equipes de front e back end, além das respostas dos colaboradores ao questionário aplicado.

### 4.1 CONTEXTO DA EMPRESA

Esta seção descreve o contexto da empresa analisada, destacando suas características organizacionais com o uso da metodologia Scrum. O objetivo é compreender a estrutura das equipes de desenvolvimento de software.

#### 4.1.1 Mapeamento das Funções com Scrum da Empresa

O mapeamento de processos “na empresa X” de base tecnológica busca identificar as atividades fundamentais que compõem o fluxo de trabalho da organização. A empresa se caracteriza como uma intermediadora de produtos digitais e físicos, conectando produtores e consumidores. Além disso, atua na interface entre os clientes e as adquirentes, oferecendo soluções tecnológicas que facilitam as transações financeiras e garantem a segurança das operações.

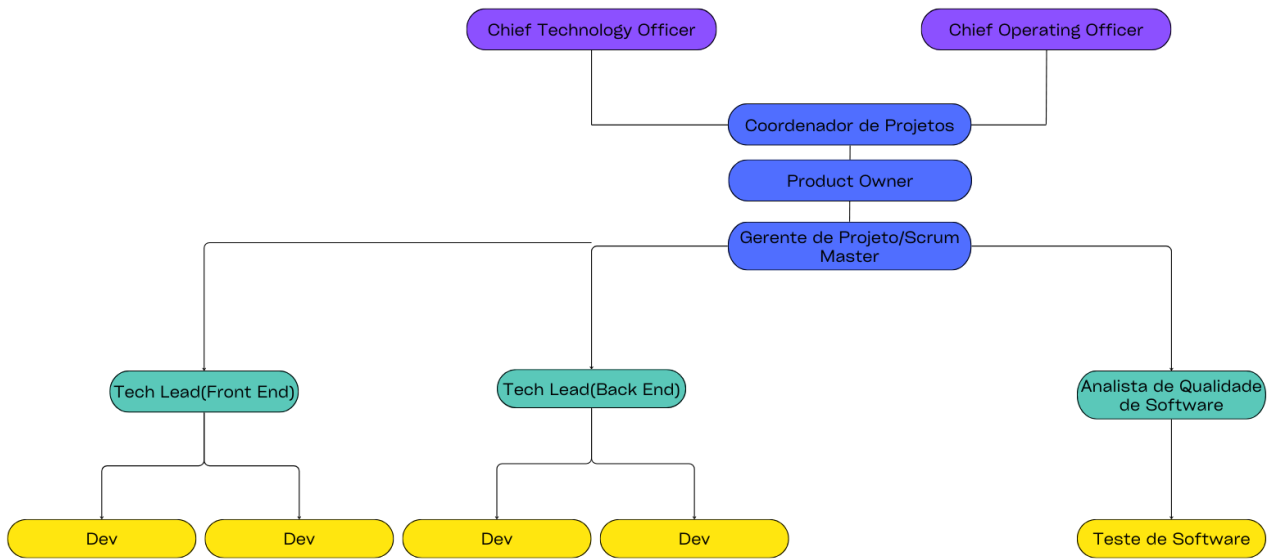
A estrutura organizacional da empresa é projetada para promover a integração entre setores essenciais. Cada setor desempenha um papel estratégico no cumprimento dos objetivos corporativos. Em relação à área da tecnologia, integram-se os setores de tecnologia e operações, e a estrutura hierárquica apresentada no esquema da Figura 4 organiza uma equipe de tecnologia, sendo composta por diferentes níveis de responsabilidade e funções inter-relacionadas. No topo da hierarquia, encontra-se o Chief Technology Officer (CTO), responsável por definir a visão tecnológica da organização, supervisionar a implementação de tecnologias e alinhar os recursos de TI aos objetivos estratégicos do negócio. Paralelamente, o Chief Operating Officer (COO) atua na supervisão das operações

diárias, promovendo a eficiência operacional e garantindo a entrega de projetos por meio da área da qualidade de software.

No nível intermediário, está o Gerente de Projeto/Scrum Master, que desempenha um papel central na coordenação das equipes. Este profissional facilita as cerimônias do Scrum, como planejamento de sprints e retrospectivas, remove impedimentos e assegura a comunicação eficiente entre as equipes de desenvolvimento e qualidade. Abaixo dele, encontram-se os Tech Leads, divididos em Front End e Back End, que lideram tecnicamente suas respectivas equipes. O Tech Lead de Front End é responsável por definir padrões de desenvolvimento de interfaces, revisar códigos e garantir a funcionalidade e responsividade do design. Por sua vez, o Tech Lead de Back End estabelece padrões para desenvolvimento de sistemas, assegura escalabilidade e segurança, e apoia tecnicamente os membros da equipe.

Subordinados aos Tech Leads, os desenvolvedores têm como função implementar funcionalidades, corrigir erros e realizar testes básicos antes da revisão de código. Essa camada executa diretamente as demandas técnicas seguindo as diretrizes estabelecidas pela liderança. Paralelamente, o Analista de Qualidade de Software lidera os esforços de garantia de qualidade de software, planejando cenários de teste, automatizando processos quando necessário e identificando falhas para assegurar que o produto entregue atenda aos padrões esperados. Sob a orientação do analista, o testador de software realiza testes funcionais e de regressão, documenta falhas e validam o funcionamento correto das funcionalidades desenvolvidas.

Essa estrutura hierárquica promove uma organização eficiente, caracterizada pela distribuição clara de responsabilidades e pelo foco na colaboração entre os níveis. A adoção de práticas ágeis, como o Scrum, facilita a comunicação, reduz barreiras e assegura entregas contínuas e de qualidade, evidenciando uma abordagem estratégica alinhada às demandas do mercado tecnológico. Conforme o ilustrado na Figura 4:

**Figura 4:** Mapeamento Funções Scrum

**Fonte:** De autoria própria.

#### 4.1.2 Equipe de Desenvolvimento de Software e Metodologia Ágil

A equipe de desenvolvimento de software da empresa X é composta por dois times especializados, um voltado para o desenvolvimento de *front-end* e outro responsável pelo *back-end*. Juntos, esses grupos trabalham para criar e manter as plataformas tecnológicas que suportam os serviços oferecidos pela organização, garantindo soluções integradas e eficientes.

A liderança superior da área da tecnologia é exercida pelo Líder de TI, já a técnica de cada equipe, tanto Front End quanto Back End, é exercida por um Tech Lead, cuja função principal é supervisionar as atividades técnicas, garantindo que o trabalho esteja alinhado aos padrões de qualidade e objetivos organizacionais. Além disso, a equipe conta com o apoio de um Scrum Master, que atua como facilitador do processo, removendo impedimentos, promovendo a colaboração e assegurando a adoção consistente da metodologia ágil Scrum.

O trabalho segue ciclos organizados em sprints, com duração de duas semanas, durante os quais as atividades são planejadas, executadas, revisadas e entregues. As reuniões diárias (daily meetings) são realizadas no início do dia e têm

como objetivo alinhar o progresso das tarefas, identificar dificuldades e ajustar prioridades de maneira ágil.

Além das reuniões diárias, ocorrem sessões semanais de apresentação dos resultados, onde as entregas realizadas durante o sprint são discutidas com os stakeholders. Essas sessões são seguidas por reuniões de retrospectiva (retro), nas quais a equipe avalia o que funcionou bem, o que pode ser melhorado e define ações para aprimorar a performance nos próximos ciclos.

O planejamento de cada sprint é conduzido de forma colaborativa em reuniões específicas (Sprint Planning), que envolvem a definição e priorização das tarefas a serem realizadas, considerando a capacidade dos membros da equipe e os objetivos estratégicos da empresa.

A governança geral das atividades das equipes de desenvolvimento é orientada pelos CEOs da organização, que coordenam os esforços entre os setores e define as prioridades estratégicas. Essa integração entre liderança técnica, metodologias ágeis e governança estratégica permite que os times de desenvolvimento entreguem soluções alinhadas às demandas do mercado e às expectativas dos clientes.

## 4.2 ANÁLISE DAS MÉTRICAS

### 4.2.1. Análise Equipe Front End

Os dados utilizados para a análise foram extraídos das documentações das retrospectivas de cada sprint, registradas no Confluence, uma ferramenta que é utilizada por equipes ágeis para gerenciar conhecimento e colaborar em projetos. Além disso, o acompanhamento do progresso das sprints e a mensuração dos itens concluídos foram realizados utilizando o Jira, uma plataforma de gerenciamento de projetos e tarefas que facilita a aplicação da metodologia Scrum. Abaixo é apresentada a tabela com os dados das sprint do time *Front-End*:

**Tabela 2:** Métricas Time *Front-End*

<b>Sprint</b>	<b>Itens Concluídos</b>	<b>Itens Não Concluídos</b>	<b>Velocidade</b>
Sprint 22	17	11	22
Sprint 23	26	4	71
Sprint 24	19	10	55
Sprint 25	27	3	70
Sprint 26	20	8	47
Sprint 27	-	-	-
Sprint 28	26	6	50
Sprint 29	36	3	55
Sprint 30	25	16	48
Sprint 31	36	4	50
Sprint 32	27	2	54
Sprint 33	16	9	30
Sprint 34	36	1	59
Sprint 35	32	5	48
Sprint 36	21	30	65
Sprint 37	25	13	60
Sprint 38	48	5	76

**Fonte:** De autoria própria.

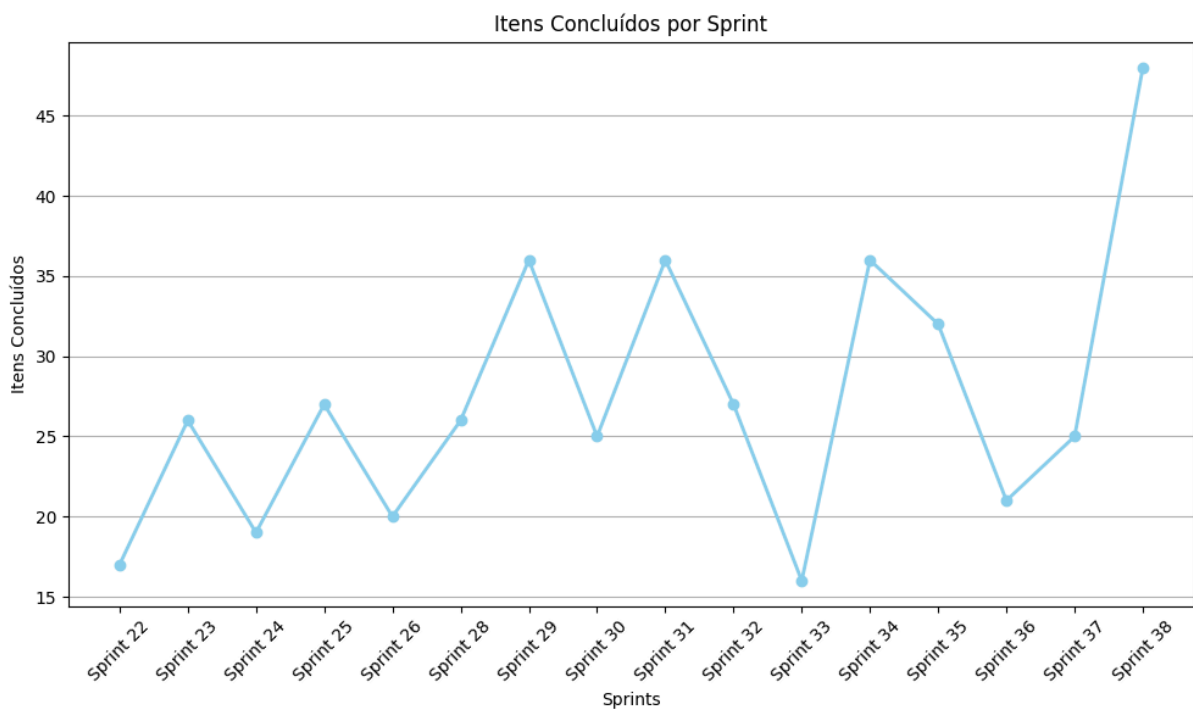
A partir dos dados coletados, foi possível calcular estatísticas descritivas que fornecem uma visão do desempenho da equipe. As estatísticas descritivas são fundamentais para compreender a distribuição e a variabilidade dos dados . Nesse sentido, na tabela 3 são apresentados os resultados da análise estatística dos dados das sprints do time *Front-End*:

**Tabela 3:** Estatística Descritiva

Estatística	Itens Concluídos	Itens Concluídos	Velocidade
Média	27.3	8.1	53.75
Desvio Padrão	8.5	7.2	13.61
Mínimo	16.0	1.0	22
Máximo	48.0	30.0	76
Mediana	26.0	5.5	54.5

**Fonte:** De autoria própria.

A Figura 5 é um gráfico do número de itens concluídos pela equipe de *Front-End* em cada sprint analisada. Ele evidencia o desempenho da equipe ao longo do tempo, permitindo identificar padrões, picos de produtividade e possíveis variações no volume de entregas.

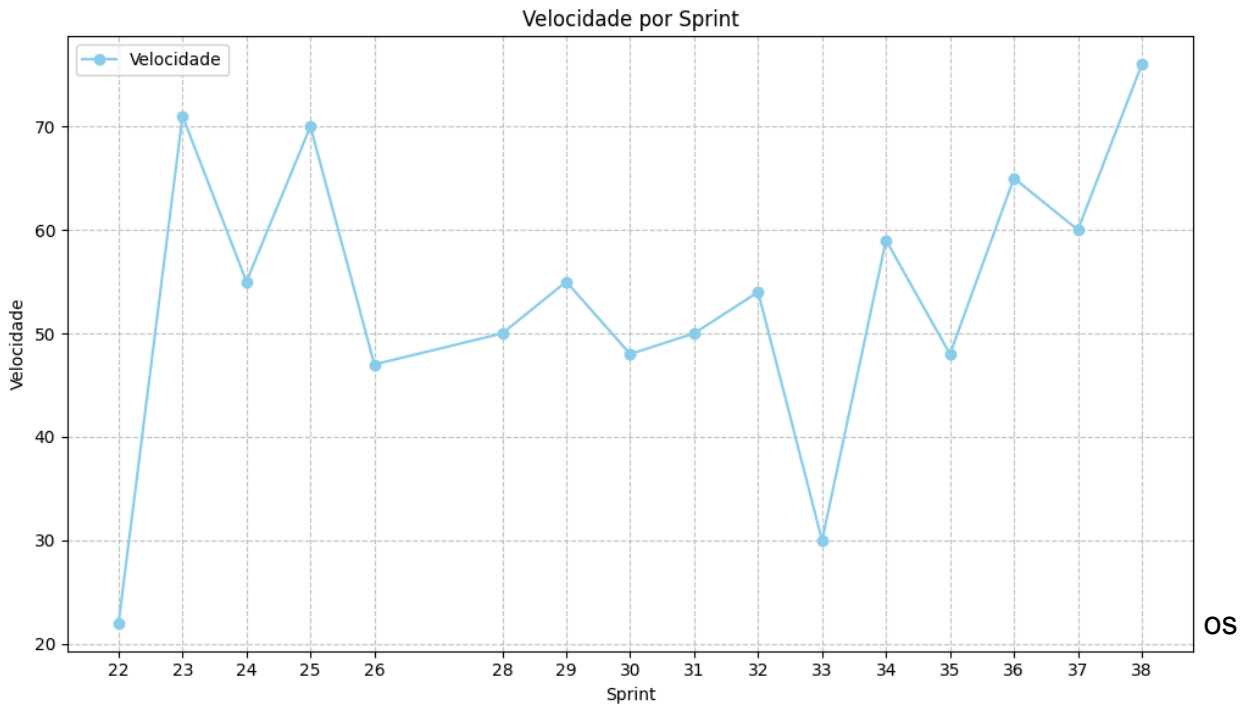
**Figura 5:** Gráfico itens concluídos por sprint

**Fonte:** De autoria própria.

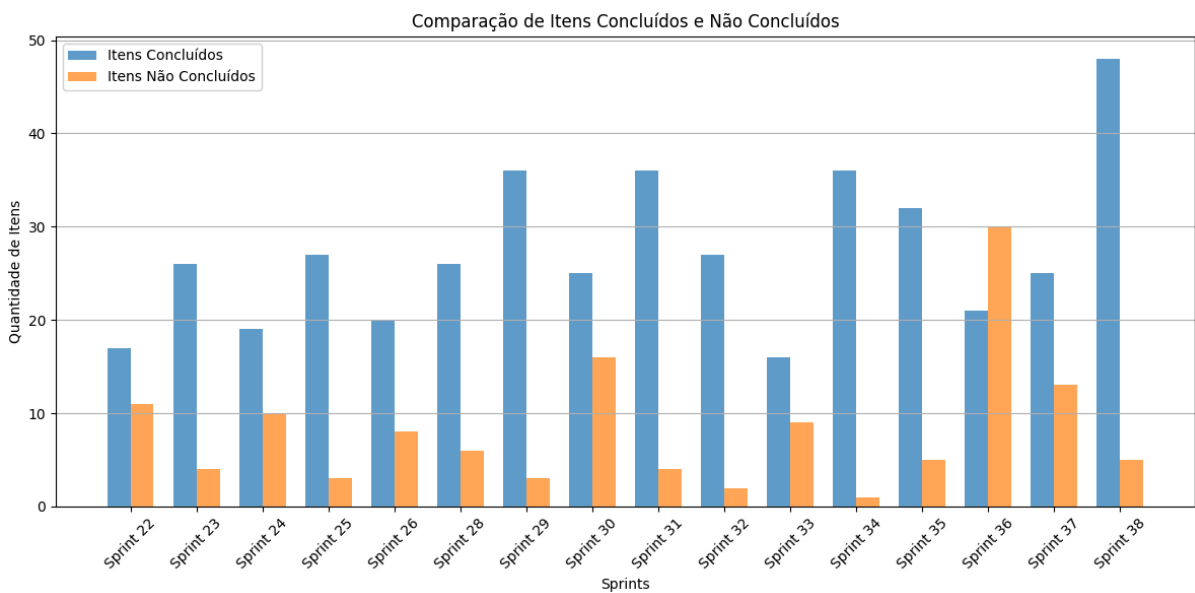
O gráfico da Figura 6 corresponde à velocidade por sprint e ilustra a capacidade da equipe de entregar valor ao final de cada sprint, representada pela

soma dos pontos de esforço associados aos itens concluídos. Este indicador é essencial para o planejamento e a previsibilidade de futuros ciclos, além de servir como referência para a análise da evolução da equipe ao longo do tempo.

**Figura 6:** Gráfico velocidade por sprint



**Figura 7:** Gráfico de barras itens concluídos e não concluídos



**Fonte:** De autoria própria.

A análise dos dados de desempenho da equipe de *front-end* ao longo de 16 sprints revela resultados positivos, mas com pontos de atenção. A média de itens concluídos foi de 27,3, enquanto a mediana foi de 26, indicando consistência na produtividade, embora com uma leve assimetria devido a valores altos, como os 48 itens concluídos na Sprint 38. A média de itens não concluídos foi de 8,1, mas a Sprint 36 apresentou um desempenho atípico, com 30 itens não concluídos, evidenciando possíveis problemas de planejamento ou mudanças de escopo. A velocidade média da equipe foi de 53,75, com um desvio padrão de 13,61, refletindo variações nas condições de trabalho e na complexidade das sprints.

A relação entre velocidade, itens concluídos e itens não concluídos destaca padrões importantes. A maior velocidade (76) foi registrada na Sprint 38, que também teve o maior número de itens concluídos (48), sugerindo eficiência na execução das tarefas. Contudo, a Sprint 36, apesar de uma velocidade relativamente alta (65), apresentou um número elevado de itens não concluídos, indicando que velocidade elevada nem sempre reflete maior eficiência. Esses resultados apontam para a importância de equilibrar a carga de trabalho e planejamento para garantir a eficiência das tarefas atribuídas à equipe.

No geral, a equipe demonstrou potencial para alcançar resultados elevados quando fatores como planejamento, estimativas de esforço e clareza nos objetivos estão alinhados. Recomenda-se a realização de retrospectivas mais detalhadas e a utilização de ferramentas de monitoramento, como gráficos de acompanhamento de progresso, para identificar e mitigar riscos durante as sprints. Assim, será possível reduzir a variabilidade e aumentar a previsibilidade, contribuindo para uma performance ainda mais consistente e eficiente.

#### 4.2.2 Análise Equipe Back End

Abaixo é apresentada a tabela com os dados das sprint do time *Back-End*:

**Tabela 4:** Métricas Time *Back-End*

<b>Sprint</b>	<b>Itens Concluídos</b>	<b>Itens Não Concluídos</b>	<b>Velocidade</b>
Sprint 22	19	13	27
Sprint 23	19	11	26
Sprint 24	3	22	1
Sprint 25	28	5	51
Sprint 26	13	6	55
Sprint 27	23	3	52
Sprint 28	20	4	42
Sprint 29	15	7	41
Sprint 30	17	2	51
Sprint 31	14	5	34
Sprint 32	27	2	50
Sprint 33	17	10	41
Sprint 34	-	-	-
Sprint 35	14	14	36
Sprint 36	20	23	38
Sprint 37	24	6	57
Sprint 38	24	6	52

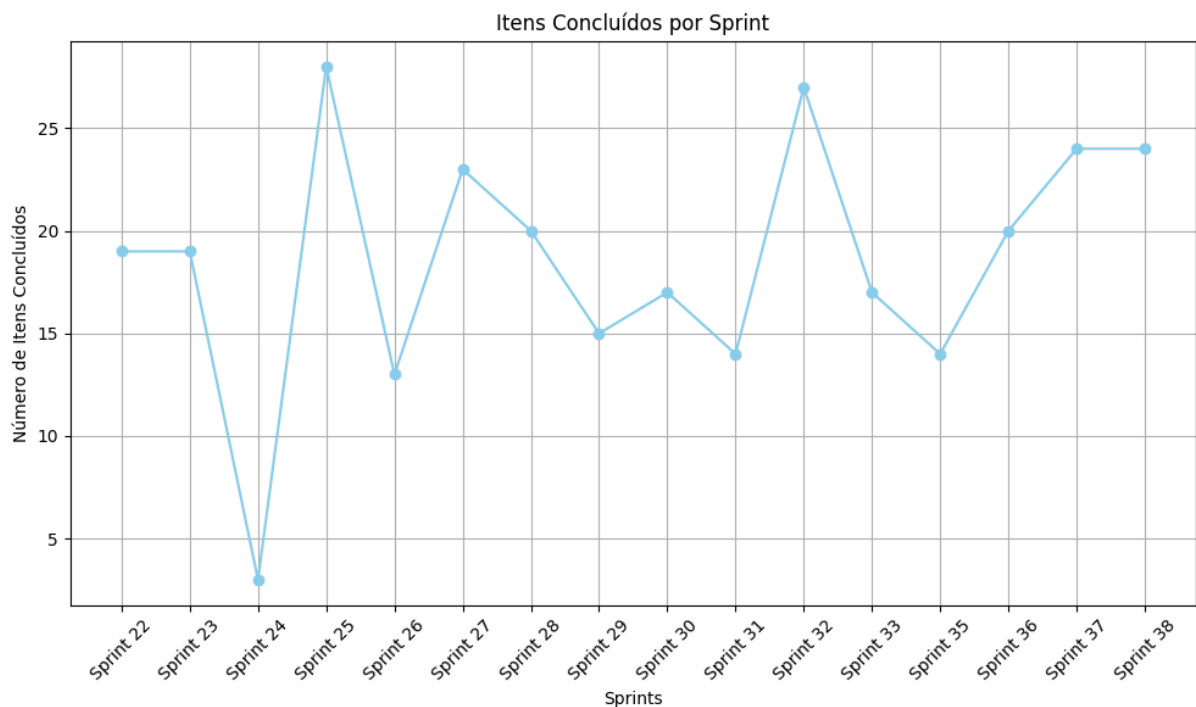
**Fonte:** De autoria própria.

**Tabela 5:** Análise Descritiva *Back-End*

Estatística	Itens Concluídos	Itens Não Concluídos	Velocidade
Média	18.5	8.7	40.88
Desvio	6.2	6.48	13.87
Mínimo	3.0	2.0	1
Máximo	28.0	23.0	57
Mediana	19.0	6.0	41.5

**Fonte:** De autoria própria.

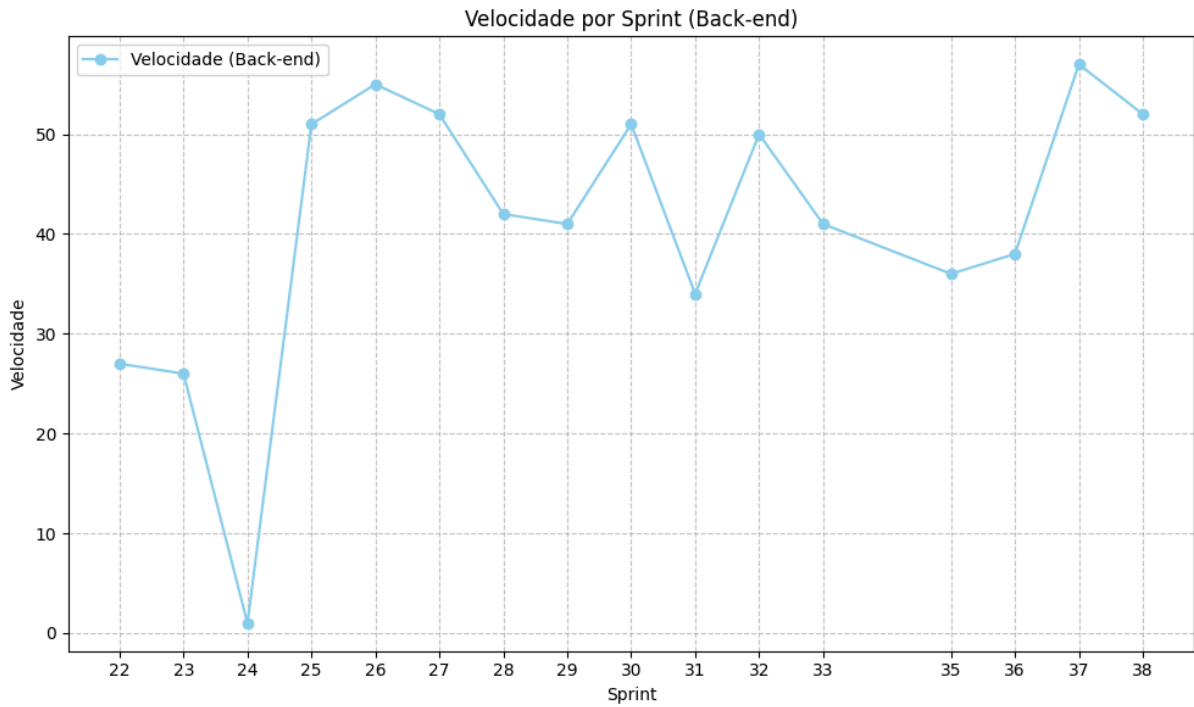
O gráfico da Figura 8 apresenta o número de itens concluídos pela equipe *Back-End* em cada sprint, permitindo visualizar tendências de produtividade e identificar variações ao longo do tempo.

**Figura 8:** Gráfico itens concluídos por sprint

**Fonte:** De autoria própria.

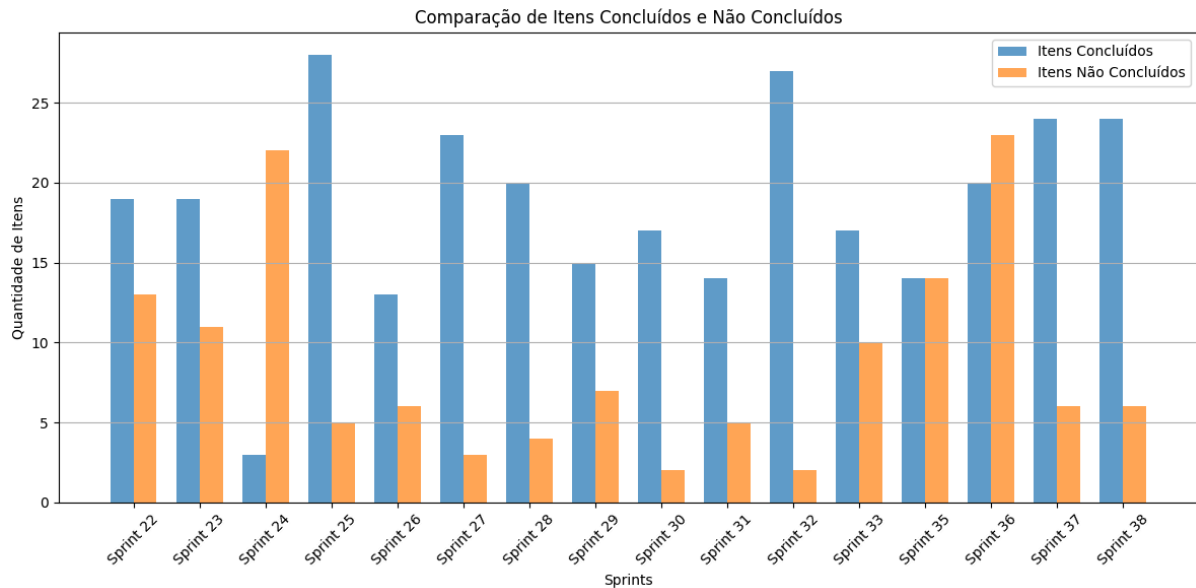
A Figura 9 representa um gráfico que demonstra a velocidade da equipe, medida em pontos entregues por sprint. Ele fornece uma visão sobre a capacidade de entrega e a consistência do desempenho da equipe.

**Figura 9:** Gráfico velocidade por sprint



**Fonte:** De autoria própria.

A Figura 10 compara os itens concluídos e não concluídos por sprint, revelando a proporção entre o entregue e o não entregue.

**Figura 10:** Gráfico de barra itens concluídos e itens não concluídos

**Fonte:** De autoria própria.

A análise dos dados do time de back end ao longo de 16 sprints evidencia um desempenho geral estável, mas com variações significativas que apontam áreas de melhoria. A equipe concluiu, em média, 18,5 itens por sprint, com uma mediana de 19 itens, indicando uma produtividade relativamente consistente, embora tenha enfrentado situações atípicas, como a Sprint 24, na qual apenas 3 itens foram concluídos. Esse resultado sugere possíveis problemas de planejamento ou mudanças no escopo. Por outro lado, sprints como a 25 e a 32 demonstraram alta produtividade, com 28 e 27 itens concluídos, respectivamente, e baixo número de itens não concluídos, associados a velocidades elevadas de 51 e 50.

A média de itens não concluídos foi de 8,7, com um desvio padrão de 6,48, refletindo maior variabilidade nessa métrica em comparação aos itens concluídos. A velocidade média foi de 40,88, com desvio padrão de 13,87, indicando diferenças significativas entre as sprints. A relação entre a velocidade e os itens concluídos mostra que, embora uma maior velocidade geralmente esteja associada a maior produtividade, isso nem sempre é consistente. Por exemplo, na Sprint 36, a equipe concluiu 20 itens e deixou 23 pendentes, mesmo com uma velocidade intermediária de 38, evidenciando desafios na execução ou um escopo mal definido.

Os resultados mais consistentes ocorreram nas sprints finais, como a 37 e a 38, nas quais foram concluídos 24 itens com apenas 6 não concluídos e velocidades

elevadas de 57 e 52. Isso sugere que o time conseguiu melhorar o equilíbrio entre produtividade e controle de tarefas pendentes, especialmente ao final do período analisado. Contudo, momentos de desempenho irregular, como a Sprint 24, destacam a necessidade de ajustes no planejamento e na gestão do escopo.

De forma geral, o desempenho do time demonstra potencial para alcançar alta produtividade quando planejamento, execução e escopo estão bem alinhados. Recomenda-se a análise detalhada das sprints com melhores resultados para replicar as práticas bem-sucedidas, assim como o aprofundamento nas retrospectivas das sprints mais problemáticas para identificar e mitigar os fatores que dificultaram a execução das tarefas. Estratégias como o uso de ferramentas de acompanhamento, estimativas mais precisas e maior estabilidade no ritmo de trabalho podem contribuir para reduzir variações e garantir resultados mais previsíveis e eficientes.

#### 4.3 QUESTIONÁRIO

O questionário foi elaborado com o objetivo de avaliar as percepções, preferências e experiências dos respondentes em relação ao uso de metodologias ágeis no contexto de desenvolvimento de projetos de software. A estrutura das perguntas foi cuidadosamente planejada para conduzir o respondente por uma reflexão gradual, que permite não apenas identificar a familiaridade com essas metodologias, mas também comparar sua aplicabilidade em relação às abordagens tradicionais.

A primeira questão solicita ao respondente que indique sua preferência entre metodologias ágeis, tradicionais ou híbridas. No entanto, ao longo do questionário, são feitas perguntas específicas e contextualizadas que exploram experiências, opiniões e percepções sobre aspectos fundamentais da metodologia ágil, como comunicação, entrega contínua de valor, desafios enfrentados e organização do trabalho.

Essa abordagem foi intencionalmente planejada para identificar possíveis contradições nas respostas, caso o respondente afirme inicialmente conhecer ou preferir a metodologia ágil, mas demonstre, por meio das respostas subsequentes, desconhecimento de conceitos ou dificuldades em aplicá-los. Essa técnica permite

uma análise mais aprofundada, validando se a mesma está alinhada com a prática e a compreensão real da metodologia.

Ao adotar esse formato, o questionário busca não apenas coletar dados sobre preferências, mas também avaliar o nível de familiaridade e consistência dos respondentes, contribuindo para uma análise crítica sobre a adoção de metodologias ágeis no ambiente organizacional.

Os resultados da pesquisa indicam uma predominância da preferência pela metodologia ágil, entre os respondentes. Todos os 7 participantes manifestaram optar por essa abordagem, destacando a flexibilidade e a interatividade como principais razões. No que tange à clareza dos papéis e responsabilidades nas equipes ágeis, a percepção geral é positiva, sendo considerada "muito clara" por 43% dos respondentes e "clara" por 57%. Esses dados sugerem que, quando bem implementada, a metodologia ágil promove organização e definição de papéis dentro das equipes.

Os principais benefícios associados à adoção da metodologia ágil incluem a melhoria da comunicação (85,7%), a redução do tempo de entrega (86%), o aumento da motivação da equipe (57%) e a maior satisfação do cliente (43%). Esses resultados corroboram estudos prévios que apontam as metodologias ágeis como facilitadoras de maior colaboração e entregas eficientes. Por outro lado, foram apontadas dificuldades como restrições de tempo e recursos (67%), dificuldade em adaptar as práticas ao contexto organizacional (50%) e resistência cultural (17%), evidenciando os desafios da implementação em certos ambientes.

Quanto à viabilidade da metodologia ágil em projetos que requerem maior previsibilidade, a maioria dos respondentes (71%) considera que ela é aplicável com adaptações, enquanto 29% acredita em sua viabilidade sem necessidade de mudanças. Esses dados reforçam a flexibilidade das práticas ágeis, permitindo seu uso em contextos que exigem maior controle e previsibilidade.

Em relação à organização do trabalho, 43% dos participantes relataram atuar de forma mista, alternando entre estruturação e adaptação, enquanto 29% optaram por abordagens mais flexíveis, ajustando-se conforme as necessidades do projeto, e outros 29% preferem métodos estruturados, com planejamento detalhado e prazos fixos. Essa distribuição indica uma busca por equilíbrio entre rigor e adaptabilidade, especialmente em cenários que demandam maior previsibilidade.

A preferência no planejamento de novos projetos reflete essa visão equilibrada: 86% dos participantes preferem processos que conciliam planejamento com flexibilidade para mudanças, enquanto 14% priorizam ajustes contínuos ao longo do desenvolvimento. Similarmente, a comunicação no desenvolvimento dos projetos é majoritariamente vista como ideal quando combina reuniões frequentes com períodos de foco individual (86%), ressaltando a importância tanto da colaboração quanto da produtividade individual.

No que diz respeito à definição de sucesso em projetos, 43% dos respondentes consideram uma combinação entre cumprimento de metas e flexibilidade para melhorias como a principal métrica, enquanto 43% priorizam a entrega de um produto de qualidade que atenda às necessidades, mesmo com ajustes. Apenas 14% dos participantes definem sucesso estritamente como atingir os objetivos dentro do prazo e orçamento planejados.

Em relação a comunicação durante o desenvolvimento de um projeto, 71% prefere uma combinação entre reuniões frequentes e períodos de foco individual. Já 29% optou por reuniões mais espaçadas e foco na execução individual.

Em relação ao alcance dos resultados, 57% preferem ser capazes de adaptar-se às mudanças durante o processo, garantindo a entrega final, 14% cumprir rigorosamente o planejamento e prazos definidos. Enquanto que 29% prefere uma combinação entre os dois.

Sendo assim, em geral, os dados analisados apontam para uma preferência consolidada pela metodologia ágil, principalmente devido aos benefícios associados à flexibilidade e à eficiência. Contudo, as respostas apontadas revelam a necessidade de adaptações para sua aplicação em contextos específicos, sugerindo que abordagens híbridas podem ser alternativas viáveis para equilibrar planejamento e adaptabilidade em diferentes tipos de projetos.

#### **4.3.1. Análise do questionário**

O questionário foi respondido por 7 participantes, fornecendo uma visão geral sobre a adoção da metodologia ágil no ambiente de trabalho, com uma clara preferência dos participantes por práticas ágeis. A alta avaliação da preferência das metodologias ágeis é observada em 100% das respostas.

Os benefícios apontados, como a melhoria da comunicação, redução do tempo de entrega, aumento da motivação da equipe e maior satisfação do cliente, corroboram as vantagens amplamente reconhecidas das metodologias ágeis. A melhoria da comunicação e a redução do tempo de entrega são frequentemente destacadas como efeitos imediatos das práticas ágeis, devido à ênfase na interação constante entre os membros da equipe e o ciclo de entregas rápidas. Além disso, a motivação da equipe é frequentemente observada em ambientes ágeis devido ao foco na autonomia e no empoderamento dos membros, o que também se reflete nos resultados da pesquisa.

Contudo, as dificuldades identificadas, como as restrições de tempo e recursos, a resistência cultural e a adaptação das práticas ágeis ao contexto da empresa, indicam que, embora a metodologia ágil seja eficaz em muitos casos, sua implementação nem sempre é isenta de desafios. Esses desafios são amplamente reconhecidos na literatura, que aponta a resistência cultural como uma das barreiras mais comuns à adoção de práticas ágeis (Fitriani et al., 2024). Além disso, a dificuldade em adaptar as práticas ágeis ao contexto organizacional sugere que a metodologia ágil não é uma solução universal e pode exigir ajustes para se adequar às especificidades de cada ambiente de trabalho.

A percepção de que a metodologia ágil pode ser aplicada com adaptações em projetos que exigem maior previsibilidade (71% dos respondentes) indica uma flexibilidade reconhecida na abordagem, mas também aponta para a necessidade de ajustes dependendo do tipo de projeto. A metodologia ágil, frequentemente associada a projetos mais dinâmicos e imprevisíveis, tem sido cada vez mais vista como uma abordagem viável para projetos que exigem maior controle e previsibilidade, especialmente quando combinada com práticas híbridas que equilibram a flexibilidade com uma estruturação mais rígida.

As diferentes formas de organização de trabalho e o planejamento de novos projetos, como evidenciado pelos 43% que adotam uma abordagem mista, sugerem que a flexibilidade é vista como essencial para responder às mudanças, mas também há uma valorização de uma estrutura planejada, especialmente quando surgem desafios.

Os resultados indicam que, embora a metodologia ágil seja amplamente preferida, a necessidade de adaptar suas práticas ao contexto de cada organização

e tipo de projeto é evidente. Isso reforça a importância de abordagens híbridas, que permitem combinar a flexibilidade das metodologias ágeis com a estruturação de métodos tradicionais, para alcançar um equilíbrio entre adaptação às mudanças e cumprimento de objetivos bem definidos. Em suma, os resultados da pesquisa ressaltam a eficácia da metodologia ágil, mas também apontam para os desafios de sua aplicação universal, sugerindo que abordagens mistas podem ser a solução mais adequada para equilibrar flexibilidade e previsibilidade em diferentes contextos de projetos.

## 5. ANÁLISE COMPARATIVA DOS RESULTADOS

### 5.1 COMPARATIVO METODOLOGIA SCRUM E ÁGIL

A fundamentação teórica do trabalho aborda os conceitos de metodologias ágeis, com foco no framework Scrum, que é utilizado para promover a organização e a eficiência no desenvolvimento de software. A prática do Scrum, conforme apresentada na teoria, busca uma maior integração entre as equipes, promovendo a colaboração constante e entregas incrementais e contínuas. A empresa X, como uma intermediadora de produtos digitais e físicos, adota o Scrum em suas equipes de tecnologia e operações para atender às necessidades do mercado de forma ágil, integrando diferentes setores da organização, como tecnologia, marketing e operações.

### 5.2 USO DO SCRUM NA EMPRESA

De acordo com a fundamentação teórica, o Scrum organiza o trabalho em ciclos iterativos denominados sprints, com foco em entregas incrementais a cada ciclo (Schwaber e Beedle, 2002). A empresa X segue essa estrutura básica do Scrum, aplicando o conceito de sprints para organizar os fluxos de trabalho. No entanto, a prática demonstrou que a empresa adaptou certos aspectos do Scrum para atender à dinâmica do seu ambiente de negócios.

A teoria do Scrum estabelece que o Product Owner (PO) deve ser o responsável pelo gerenciamento do backlog do produto, ou seja, a priorização das funcionalidades a serem desenvolvidas com base nas necessidades do cliente e do mercado. (Rubin, 2017) Na empresa X, o PO desempenha essa função de forma bastante alinhada à teoria, mas com algumas adaptações. O PO não apenas gerencia o backlog, mas também tem um papel ativo na área de User Interface e User Experience interagindo diretamente como time de front end.

Entretanto, uma adaptação importante observada na prática da empresa X é que, além de priorizar as funcionalidades no backlog, o PO também auxilia em questões operacionais que, na teoria, seriam de responsabilidade de outros setores. Isso ocorre porque, na empresa X, o PO se envolve diretamente no gerenciamento

de processos que envolvem a interface entre os diferentes setores da organização, especialmente entre tecnologia e operações.

Além disso, na empresa X, essa integração entre setores é observada na prática, especialmente entre as equipes de tecnologia e operações. Durante os sprints, as equipes de desenvolvimento de software trabalham diretamente com o time de operações para garantir que as entregas atendam às expectativas dos clientes e que o fluxo de trabalho esteja alinhado às necessidades do mercado.

### 5.3 COMPARATIVO POR TIPO DE PROJETO(*FRONT-END* E *BACK-END*)

A análise comparativa da performance dos times de desenvolvimento, especificamente entre as equipes de frontend e backend, é fundamental para entender a eficiência dos processos de entrega de software. Vale destacar que um ponto que pode interferir na produtividade da equipe é a ausência de uma documentação bem estabelecida, fato que ocorre no time de back end.

O time de *front-end* demonstrou uma produtividade superior, com uma média de 27,3 itens concluídos por sprint, comparada aos 18,5 do back end. A mediana de itens concluídos também reflete essa diferença: 26 para o *front-end* contra 19 para o back end. O desvio padrão para itens concluídos foi maior no front end (8,5) em relação ao *back-end* (6,2), indicando maior variabilidade nos resultados do front end, possivelmente devido a mudanças no escopo ou na complexidade das tarefas.

Ambos os times apresentaram uma média similar de itens não concluídos: 8,1 no front end e 8,7 no *back-end*. No entanto, o desvio padrão foi ligeiramente maior no back end (6,48) do que no *front-end* (7,2), refletindo maior instabilidade no cumprimento das tarefas planejadas pelo back end. O número máximo de itens não concluídos foi mais elevado no *back-end* (23 na Sprint 36) em comparação ao *front-end* (30 na mesma sprint), sugerindo que, apesar de desafios pontuais, o *front-end* teve maior capacidade de recuperar desempenho em sprints subsequentes.

A velocidade média foi maior no front end (53,75) em relação ao *back-end* (40,88). Essa diferença reflete maior agilidade do time de front end em executar suas tarefas. Contudo, a velocidade do *back-end* apresentou maior estabilidade

(desvio padrão de 13,87) em comparação ao front end (13,61), indicando que o front end enfrentou maior variabilidade em seu desempenho.

Ambos os times enfrentaram sprints desafiadoras: o *back-end* na Sprint 24, com apenas 3 itens concluídos e 22 não concluídos, e o *front-end* na Sprint 36, com 21 itens concluídos e 30 não concluídos. No entanto, o front end apresentou maior recuperação nas Sprints seguintes, como demonstrado pelo desempenho excepcional na Sprint 38, com 48 itens concluídos e velocidade de 76, enquanto o *back-end* manteve resultados mais modestos mesmo em suas melhores sprints.

O time de *front-end* se destacou pela capacidade de concluir um maior número de itens e alcançar velocidades elevadas, mas apresentou maior variabilidade no desempenho. O *back-end*, por sua vez, teve desempenho mais estável, mas sua menor produtividade sugere que melhorias na definição de tarefas e no planejamento poderiam aumentar sua eficiência. Ambos os times enfrentaram dificuldades em momentos específicos, destacando a importância de retrospectivas detalhadas para identificar e mitigar os fatores que levaram aos resultados abaixo do esperado.

Uma ressalva importante é que, enquanto o time de Front End dispunha de documentação detalhada para orientar o desenvolvimento, o time de Back End trabalhou sem uma documentação clara, o que pode ter contribuído para a menor produtividade observada. Esse cenário parece contrastar com o que é proposto pelo Manifesto Ágil, que prioriza "software em funcionamento sobre documentação abrangente" (BECK et al., 2001). No entanto, embora a falta de documentação no Back End possa ter impactado o desempenho, não é possível afirmar com certeza que este foi o único fator responsável pela diferença na produtividade entre os times.

Conclui-se que, para ambos os times, é essencial um maior alinhamento no planejamento das sprints, com prioridade para a clareza das tarefas e a definição de metas realistas. O time de *front-end* deve concentrar esforços na redução da variabilidade em seu desempenho, enquanto o time de *back-end* necessita implementar estratégias para aumentar sua produtividade, minimizar a ocorrência de itens não concluídos e, sobretudo, investir na criação e manutenção de uma documentação adequada. A ausência de documentação no time de back end pode estar contribuindo para inconsistências e dificuldades na execução das tarefas, além de prejudicar o aprendizado contínuo e a repetição de boas práticas. Dessa forma,

espera-se que ambos os times consigam alcançar um desempenho mais consistente e eficiente ao longo dos Sprints futuros.

## 6. CONCLUSÃO

O estudo realizado evidencia a predominância das metodologias ágeis no desenvolvimento de software, refletindo sua capacidade de atender às demandas dinâmicas e complexas do cenário atual. Contudo, os dados analisados apontam para a relevância de abordagens híbridas, que combinam a flexibilidade das metodologias ágeis com a estruturação dos métodos tradicionais. Essa estratégia pode oferecer um equilíbrio essencial entre planejamento e adaptabilidade, sendo particularmente eficaz em projetos que exigem maior previsibilidade e controle.

A análise comparativa entre as equipes de *front-end* e *back-end* revelou diferenças significativas em termos de produtividade e consistência no desempenho. No caso do time de *back-end*, a ausência de uma documentação, que é o diferencial entre as duas equipes, pode estar associada a uma maior variabilidade nos resultados, indicando que investimentos na qualidade e na padronização da documentação podem trazer ganhos substanciais.

As implicações deste estudo oferecem contribuições para organizações que buscam otimizar seus processos de desenvolvimento de software. A adoção de metodologias ágeis e híbridas, quando ajustada às características específicas de cada equipe e projeto, pode melhorar a colaboração, a entrega contínua de valor e a satisfação do cliente.

Por fim, sugere-se que pesquisas futuras analisem mais detalhadamente o impacto da documentação no desenvolvimento de software. Investigar como a ausência ou a insuficiência de documentação influencia o desempenho, especialmente em equipes técnicas como *back-end*, pode fornecer outras visões para a formulação de estratégias que equilibrem agilidade e qualidade nos processos. Este tipo de análise não apenas complementaria os achados deste estudo, mas também contribuiria para o desenvolvimento de modelos mais eficazes de gestão de projetos em diferentes contextos organizacionais.

## REFERÊNCIAS

AMBLER, Scott W. **Modelagem ágil: práticas eficazes para a programação extrema e o processo unificado**. Porto Alegre: Bookman, 2004.

ALAM, Sehrish; et al. Impact and Challenges of Requirement Engineering in Agile Methodologies: **A Systematic Review**. **International Journal of Advanced Computer Science and Applications**, v. 8, 2017.

BECK, K. E. A. **Manifesto for Agile Software Development**. 2001. Disponível em: <http://agilemanifesto.org/>. Acesso em: out. 2024.

CONFORTO, Edmilson Carlos; AMARAL, Daniel Capaldo; DA SILVA, Sérgio Luis; DI FELIPPO, Alexandre; KAMIKAWACHI, Débora Sayuri Leal. The agility construct on project management theory. **International Journal of Project Management**, v. 34, p. 660-674, 2016. DOI: 10.1016/j.ijproman.2016.01.007.

DERBY, E.; LARSEN, Diana. **Agile retrospectives: making good teams great**. Raleigh, NC: Pragmatic Programmers, 2006.

DICIONARIOTEC. Modelo cascata. **Dicionariotec**, 2022. Disponível em: <https://dicionariotec.com/posts/modelo-cascata>. Acesso em: 3 dez. 2024.

DINGSØYR, T.; NERUR, S.; BALIJEPALLY, V.; MOE, N. B. A decade of agile methodologies: Towards explaining agile software development. **Journal of Systems and Software**, v. 85, n. 6, p. 1213–1221, 2012. DOI: 10.1016/j.jss.2012.02.033.

FERREIRA, Jennifer; NOBLE, James; BIDDLE, Robert. Agile development iterations and UI design. In: **AGILE CONFERENCE**, 2007, Washington, DC. Proceedings [...]. IEEE, 2007. Disponível em: <https://www.researchgate.net/publication/4270500>. Acesso em: 2024.

FITRIANI, Widia Resti; RAHAYU, Puji; SENSUSE, Dana Indra. **Challenges in agile software development: a systematic literature review**. In: **International Conference on Advanced Computer Science and Information Systems**

(**ICACISIS**), 2016, Jakarta, Indonésia. Anais [...]. Jakarta: Universitas Indonesia, 2016.

GARCÍA-MORENO, C.; MARTÍNEZ-GARCÍA, M. J. Estudo de caso como abordagem de pesquisa na gestão de projetos. **Revista de Gestão, Liderança e Tecnologia**, v. 1, n. 2, p. 99-110, 2012.

HAYAT, Faisal; REHMAN, Ammar Ur; ARIF, Khawaja Sarmad; WAHAB, Kanwal; ABBAS, Muhammad. The influence of agile methodology (Scrum) on software project management. In: **SNPD 2019: International Conference on Software, Networking and Processing Devices**, Toyama, Japão, 8-11 jul. 2019. IEEE, 2019.

HIGHSMITH, J. **Agile software development ecosystems**. Boston, MA: Addison-Wesley, 2002.

HOSSAIN, E.; BABAR, M. A.; PAIK, H. Y. Using Scrum in global software development: a systematic literature review. In: **ICGSE 2009: Fourth IEEE International Conference on Global Software Engineering, Proceedings**. IEEE Comp Sci, 2009. DOI: 10.1109/ICGSE.2009.25.

LAPKIN, A. et al. Gartner clarifies the definition of the term 'enterprise architecture'. 2008.

LARSON, D.; CHANG, V. A review and future direction of agile, business intelligence, analytics and data science. **International Journal of Information Management**, v. 36, n. 5, p. 700–710, 2016. DOI: 10.1016/j.ijinfomgt.2016.04.013.

LÓPEZ-MARTÍNEZ, Janeth et al. Problems in the adoption of agile-Scrum methodologies: a systematic literature review. In: **2016 4th International Conference in Software Engineering Research and Innovation (CONISOFT)**, San Luis Potosí, México, 2016. IEEE, 2016.

MOE, N. B.; DINGSØYR, T.; DYBÅ, T. A teamwork model for understanding an agile team: a case study of a Scrum project. **Information and Software Technology**, v. 52, n. 5, p. 480–491, 2010. DOI: 10.1016/j.infsof.2009.11.013.

MORETTIN, P. A.; BUSSAB, W. de O. **Estatística Básica**. 6. ed. São Paulo: Saraiva, 2010. ISBN 978-85-02-08177-2.

PEREIRA, Julio Cesar; RUSSO, Rosaria de F. S. M. Design thinking integrated in agile software development: a systematic literature review. **Procedia Computer Science**, v. 138, p. 775-782, 2018. Disponível em: <https://www.sciencedirect.com>.

POMAR, Filipe Albero et al. Understanding sprint velocity fluctuations for improved project plans with Scrum: a case study. **Journal of Software: Evolution and Process**, v. 26, p. 776–783, 2014. Disponível em: <https://wileyonlinelibrary.com>. DOI: 10.1002/smr.1661.

PÓCSOVÁ, Jana et al. Implementation of Agile Methodologies in an Engineering Course. **Education Sciences**, Košice: Technical University of Košice, 17 nov. 2020. Disponível em: <https://www.mdpi.com/journal/education>.

PRIES-HEJE, Jan; BASKERVILLE, Richard. The translation and adaptation of agile methods: a discourse of fragmentation and articulation. 2017.

RAVAGLIA, Claudia Carrijo et al. Management of software development projects in Brazil using agile methods. **Independent Journal of Management & Production (IJM&P)**, v. 12, n. 5, July-August 2021. DOI: 10.14807/ijmp.v12i5.1353.

RIBEIRO, Afonso; DOMINGUES, Luísa. Acceptance of an agile methodology in the public sector. **Procedia Computer Science**, v. 138, p. 621-629, 2018. Disponível em: <https://www.sciencedirect.com>.

RUBIN, Kenneth S. **Scrum essencial: um guia prático do mais popular processo ágil**. 1. ed. São Paulo: Alta Books, 2017.

SCOTT, Ezequiel et al. Are learning styles useful indicators to discover how students use Scrum for the first time? **Computers in Human Behavior**, v. 36, p. 56-64, 2014.

SCHWABER, Ken; BEEDLE, Mike. *Agile Software Development with SCRUM*. Upper Saddle River, NJ: Prentice-Hall, 2001.

SCHWABER, Ken. **Agile Project Management with SCRUM**. Redmond, WA: Microsoft Press, 2004.

SINGH, Mona. U-SCRUM: An agile methodology for promoting usability. In: **Agile 2008 Conference**, 2008.

SOLINSKI, A.; PETERSEN, K. Prioritizing agile benefits and limitations in relation to practice usage. **Software Quality Journal**, v. 24, 2016. DOI: 10.1007/s11219-014-9253-3.

SRIVASTAVA, Apoorva; BHARDWAJ, Sukriti; SARASWAT, Shipra. SCRUM model for agile methodology. In: **International Conference on Computing, Communication and Automation (ICCCA 2017)**, Uttar Pradesh, Índia. Proceedings [...]. IEEE, 2017.

WERWKA, Jan; SPIECHOWICZ, Anna. Enterprise architecture approach to SCRUM processes, sprint retrospective example. **Proceedings of the Federated Conference on Computer Science and Information Systems**, v. 11, p. 1221-1228, 2017. DOI: 10.15439/2017F96.

WILLIAMS, Laurie. Agile software development methodologies and practices. **Advances in Computers**, v. 80, p. 1, 2010. DOI: 10.1016/S0065-2458(10)80001-4.

## APÊNDICE A – Questionário

### Formulário Funcionários Scrum

1. Quando se trata de gerenciar projetos, qual metodologia você prefere utilizar?

Marcar apenas uma oval.

Ágil (Scrum, Kanban, etc.)

Tradicional (Waterfall, etc.)

Híbrida (combinação de métodos)

5. Como você avalia a clareza dos papéis e responsabilidades dentro da equipe ágil?

Marcar apenas uma oval.

Muito clara

Clara

Neutra

Pouco clara

Confusa

6. Quais benefícios a metodologia ágil trouxe para sua equipe? (marque todos que se aplicam)

Marque todas que se aplicam.

Melhor comunicação

Redução do tempo de entrega

Maior satisfação do cliente

Aumento da motivação da equipe

Outro:

7. Quais dificuldades sua equipe enfrenta na aplicação da metodologia ágil? (marque todos que se aplicam)

Marque todas que se aplicam.

Falta de experiência na metodologia

Restrições de tempo e recursos

Resistência cultural

Dificuldade em adaptar as práticas ágeis ao contexto da empresa

Outro:

8. Em projetos que requerem maior previsibilidade, você considera que a metodologia ágil ainda é viável?

Marcar apenas uma oval.

Sim, com adaptações

Sim, sem adaptações

Não

9. Quando você pensa em como o trabalho é organizado na equipe, qual das opções abaixo descreve melhor sua experiência?

Marcar apenas uma oval.

Trabalho de forma estruturada, com planejamento detalhado e prazos fixos.  
Trabalho de forma mais flexível, adaptando-se conforme as necessidades do projeto.

Trabalho de forma mista, com momentos de estruturação e adaptação.

10. Ao iniciar um novo projeto, você prefere um processo de planejamento que:

Marcar apenas uma oval.

Seja mais detalhado e com prazos fixos para cada fase.

Permita ajustes ao longo do desenvolvimento, conforme surgem novas necessidades.

Equilibre planejamento com flexibilidade para mudanças.

12. Como você prefere que a comunicação dentro da equipe ocorra durante o desenvolvimento de um projeto?

Marcar apenas uma oval.

Com encontros regulares para alinhar o progresso e discutir desafios.

Com reuniões mais espaçadas e foco na execução individual.

Uma combinação entre reuniões frequentes e períodos de foco individual.

13. Como você definiria o sucesso de um projeto?

Marcar apenas uma oval.

Atingir todos os objetivos dentro do prazo e orçamento planejados.

Entregar um produto de qualidade que atenda às necessidades, mesmo que com ajustes ao longo do caminho.

Uma combinação de cumprimento de metas com flexibilidade para melhorar os resultados.

14. Em relação ao alcance de resultados, você considera mais importante:

Marcar apenas uma oval.

Cumprir rigorosamente o planejamento e prazos definidos.

Ser capaz de adaptar-se às mudanças durante o processo, garantindo a entrega final.

Uma combinação de planejamento com a capacidade de adaptação às circunstâncias